

Final Report On:

**The Geometric Conservation Based Algorithms  
for Multi-Fluid Flow at All Speeds**

**(F61775-01-W-E005, SPC 01-4005)**

Submitted to:

European Office of Aerospace Research and Development

(EOARD)

by

F. Moukalled

American University of Beirut,  
Faculty of Engineering & Architecture,  
Mechanical Engineering Department,  
P.O.Box 11-0236  
Beirut - Lebanon

Address all correspondence to Dr. F. Moukalled,

email: [memouk@aub.edu.lb](mailto:memouk@aub.edu.lb)

September 10, 2002

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
<b>1. REPORT DATE (DD-MM-YYYY)</b> 18-09-2002		<b>2. REPORT TYPE</b> Final Report		<b>3. DATES COVERED (From – To)</b> 14 August 2001 - 14-Aug-02	
<b>4. TITLE AND SUBTITLE</b>  The Geometric Conservation Based Algorithms For Multi-Fluid Flow At All Speeds			<b>5a. CONTRACT NUMBER</b> F61775-01-WE005		
			<b>5b. GRANT NUMBER</b>		
			<b>5c. PROGRAM ELEMENT NUMBER</b>		
<b>6. AUTHOR(S)</b>  Professor Fadl Moukalled			<b>5d. PROJECT NUMBER</b>		
			<b>5d. TASK NUMBER</b>		
			<b>5e. WORK UNIT NUMBER</b>		
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> American University of Beirut ME Dept. PO Box: 11-0236 Beirut Lebanon				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  N/A	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  EOARD PSC 802 BOX 14 FPO 09499-0014				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> SPC 01-4005	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b>  Approved for public release; distribution is unlimited.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b>  This report results from a contract tasking American University of Beirut as follows: The proposed work involves implementing and (thoroughly) testing the Geometric Conservation Based family of Algorithms within a structured finite-volume framework. The convection terms along the control volume faces will be evaluated using a High Resolution scheme applied within the context of the Normalized Variable Formulation methodology. In order to accelerate the convergence rate and reduce the overall computational cost of the algorithm, the outer iterations will be accelerated by using a non-linear fullmultigrid method. The discretization scheme will be second-order accurate in space and first order accurate in time (even though extension to second order accuracy should be straightforward). The newly implemented algorithms will be tested by solving a variety of two-dimensional multi-fluid flow problems in the subsonic, transonic, and supersonic regimes. Examples include: 1) Phase separation in a duct (water-air); 2) Turbulent bubbly flow in a pipe; 3) Turbulent gas-solid flow in a curved duct; 4) Dusty flow over a flat plate at subsonic flow conditions (~incompressible); 5) Dusty flow in a converging-diverging nozzle, and 6) Any problem of interest to Dr. Sekar (AFRL/PR). A detailed report describing the work will be submitted at end of contract as specified in the schedule of supplies in the submitted proposal.					
<b>15. SUBJECT TERMS</b> EOARD, Computational Fluid Dynamics (CFD), High Speed Aerodynamic, Two-Phase Flows					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b> UL	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b> Wayne Donaldson
<b>a. REPORT</b> UNCLAS	<b>b. ABSTRACT</b> UNCLAS	<b>c. THIS PAGE</b> UNCLAS			<b>19b. TELEPHONE NUMBER</b> <i>(Include area code)</i> +44 (0)20 7514 4299

## **Abstract**

This work is concerned with the implementation and testing, within a structured collocated finite-volume framework, of seven segregated algorithms for the prediction of multi-phase flow at all speeds. These algorithms belong to the Geometric Conservation Based Algorithms (GCBA) group in which the pressure correction equation is derived from the constraint equation on volume fractions (i.e. sum of volume fractions equals 1). The pressure correction schemes in these algorithms are based on SIMPLE, SIMPLER, SIMPLEC, SIMPLEX, SIMPLEM, SIMPLEST, PISO, and PRIME. Solving a variety of one- and two-dimensional laminar and turbulent two-phase flow problems in the subsonic, transonic, and supersonic regimes and comparing results with published numerical and/or experimental data assess the performance and accuracy of these algorithms. The SG method is used to solve for the one-dimensional test problems and the effects of grid size on convergence characteristics are analyzed. On the other hand, solutions for the two-dimensional problems are generated for several grid systems using the single grid method (SG), the prolongation grid method (PG), and the full non-linear multi-grid method (FMG) and their effects on convergence behavior are studied. The main outcomes of this study are the clear demonstrations of: (i) the capability of all GCBA algorithms to deal with multi-fluid flow situations; (ii) the ability of the FMG method to tackle the added non-linearity of multi-fluid flows; (iii) and the capacity of the GCBA algorithms to predict multi-fluid flow at all speeds.

## Nomenclature

$A_p^{(k)}$ , ...	coefficients in the discretized equation for $\mathbf{f}^{(k)}$ .
$B_p^{(k)}$	source term in the discretized equation for $\mathbf{f}^{(k)}$ .
$\mathbf{B}^{(k)}$	body force per unit volume of fluid/phase k.
$C_r^{(k)}$	coefficient equals to $1/R^{(k)}T^{(k)}$ .
$D_p^{(k)}[\mathbf{f}^{(k)}]$	the $D$ operator.
$H_p[\mathbf{f}^{(k)}]$	the $H$ operator.
$HP_p[\mathbf{f}^{(k)}]$	the $HP$ operator working on $\phi^{(k)}$ ( $\phi^{(k)}=u^{(k)}, v^{(k)}$ , or $w^{(k)}$ ).
$\mathbf{HP}_p[\mathbf{u}^{(k)}]$	the vector form of the $HP$ operator.
$\mathbf{I}^{(k)}$	inter-phase momentum transfer.
$\mathbf{J}_f^{(k)D}$	diffusion flux of $\mathbf{f}^{(k)}$ across cell face 'f'.
$\mathbf{J}_f^{(k)C}$	convection flux of $\mathbf{f}^{(k)}$ across cell face 'f'.
$\dot{M}^{(k)}$	mass source per unit volume.
$P$	pressure.
$\text{Pr}^{(k)}, \text{Pr}_t^{(k)}$	laminar and turbulent Prandtl number for fluid/phase k.
$\dot{q}^{(k)}$	heat generated per unit volume of fluid/phase k.
$Q^{(k)}$	general source term of fluid/phase k.
$r^{(k)}$	volume fraction of fluid/phase k.
$R^{(k)}$	gas constant for fluid/phase k.
$\mathbf{S}_f$	surface vector.
$t$	time.
$T^{(k)}$	temperature of fluid/phase k.
$U_f^{(k)}$	interface flux velocity ( $\mathbf{v}_f^{(k)} \cdot \mathbf{S}_f$ ) of fluid/phase k.
$\mathbf{u}^{(k)}$	velocity vector of fluid/phase k.
$u^{(k)}, v^{(k)}$ , ..	velocity components of fluid/phase k.
$x, y$	Cartesian coordinates.
$\ a, b\ $	the maximum of $a$ and $b$ .

## Greek Symbols

$\rho^{(k)}$	density of fluid/phase k.
$\Gamma^{(k)}$	diffusion coefficient of fluid/phase k.
$\Phi^{(k)}$	dissipation term in energy equation of fluid/phase k.
$f^{(k)}$	general scalar quantity associated with fluid/phase k.
$\mathbf{k}_f$	space vector equal to $(\hat{\mathbf{n}}_f - \gamma \hat{\mathbf{d}}_f) \mathbf{s}_f$
$\Delta_P[f^{(k)}]$	the $\Delta$ operator.
$\mathbf{m}^{(k)}, \mathbf{m}_t^{(k)}$	laminar and turbulent viscosity of fluid/phase k.
$\Omega$	cell volume.
$\beta^{(k)}$	thermal expansion coefficient for phase/fluid k.
$\delta t$	time step.

## Subscripts

e, w, .	refers to the east, west, ... face of a control volume.
E, W, ...	refers to the East, West, ... neighbors of the main grid point.
f	refers to control volume face f.
P	refers to the P grid point.

## Superscripts

C	refers to convection contribution.
D	refers to diffusion contribution.
(k)	refers to fluid/phase k.
(k)*	refers to updated value at the current iteration.
(k)°	refers to values of fluid/phase k from the previous iteration.
(k)'	refers to correction field of phase/fluid k.
old	refers to values from the previous time step.

# Introduction

The last two decades have witnessed a substantial transformation in the CFD industry; from a research means confined to research laboratories, CFD has emerged as an every day engineering tool for a wide range of industries (Aeronautics, Automobile, HVAC, etc...). This increasing dependence on CFD is due to a multitude of factors that have rendered practical the simulation of complex problems. Some of these factors are directly related to the maturity of several numerical aspects at the **core** of CFD. These include: multi-grid acceleration techniques [1-4] with enhanced equation solvers [5,6] that have decreased the computational cost of tackling large problems, better discretization techniques, unstructured grids [7-12], bounded high resolution schemes [13-18], as well as improved pressure-velocity (and density) coupling algorithms for fluid flow at all speeds [19-27]. Other factors, independent of the CFD industry, have to do with the exponential increase in processor power and decrease in microprocessor cost, whereby multiprocessors systems with large memory can now be set up at a fraction the cost of the super-computers of a decade ago.

Challenges still abound in relation to increasing the robustness of numerical techniques, improving the models used (e.g. turbulence), and extending the currently used algorithms [28-34] for the simulation of multi-phase flows at all speeds [35]. In this last area a number of algorithms have been recently reviewed and new ones proposed [36]. The basic difficulty in the simulation of multi-phase flows [36] stems from the increased algorithmic complexity that need to be addressed when dealing with multiple sets of continuity and momentum equations that are inter-coupled (interchange momentum by inter-phase mass and momentum transfer, etc.) both spatially and across fluids. Despite these complexities, successful segregated incompressible pressure-based solution algorithms have been devised. The IPSA variants devised by the Spalding Group at Imperial College [37-39] and the set of algorithms devised by the Los Alamos Scientific Laboratory (LASL) group [40-42] are examples of

incompressible multiphase algorithms. When dealing with all-speed flows, pressure-velocity-density coupling has to be accounted for. Pressure-based algorithms have been extended successfully [19-27] to account for this additional coupling.

Recently, Darwish et al. [36] extended the applicability of the available segregated single-fluid flow algorithms [35] to predict multi-fluid flow at all speeds. In their work, it was shown that the pressure correction equation can be derived either by using the geometric conservation equation or the overall mass conservation equation. Depending on which equation is used, the segregated pressure-based multi-fluid flow algorithms were classified respectively as either the Geometric Conservation Based family of Algorithms (GCBA) or the Mass Conservation Based family of Algorithms (MCBA). Moukalled et al. [43-46] implemented and tested the MCBA family and proved its capability to predict multi-fluid flow at all speeds. On the other hand, the GCBA family has not yet been implemented nor tested.

The objective of the present work is to implement and test the GCBA family within a structured finite-volume framework with the convection terms along the control volume faces evaluated using a High Resolution (HR) scheme applied within the context of the Normalized Variable and Space Formulation methodology (NVSF) [15]. To reduce the overall computational cost, the convergence rate is accelerated through the use of a non-linear full multi-grid method. The discretization scheme is second-order accurate in space and first order accurate in time.

In what follows, the governing equations are first introduced, followed by a brief description of the discretization procedure. Then the GCBA algorithms are presented, their capabilities to predict multi-fluid flow phenomena at all speeds demonstrated, and their performance characteristics (in terms of convergence history and speed) assessed. For that purpose, a total of twelve laminar and turbulent incompressible and compressible problems encompassing

dilute and dense gas-solid, and bubbly flows in the subsonic, transonic and supersonic regimes are solved. In addition, the performance of these algorithms is evaluated using (i) a single grid approach (SG), (ii) a prolongation only approach (PG) whereby the solution moves in one direction starting on the coarse grid and ending on the finest grid with the solution obtained on level  $n$  used as initial guess for the solution on level  $(n+1)$ , and (iii) finally a Full Multi-Grid (FMG) approach with a W cycle.

## The Governing Equations

In multi-phase flow the various fluids/phases coexist with different concentrations at different locations in the flow domain and move with unequal velocities. Thus, the equations governing multi-phase flows are the conservation laws of mass, momentum, and energy for each individual fluid. For turbulent multi-phase flow situations, an additional set of equations may be needed depending on the turbulence model used. These equations should be supplemented by a set of auxiliary relations. The various conservation equations needed are:

$$\frac{\partial(r^{(k)}\rho^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}) = r^{(k)}\dot{M}^{(k)} \quad (1)$$

$$\frac{\partial(r^{(k)}\rho^{(k)}\mathbf{u}^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}\mathbf{u}^{(k)}) = \nabla \cdot [r^{(k)}(\mu^{(k)} + \mu_t^{(k)})\nabla\mathbf{u}^{(k)}] + r^{(k)}(-\nabla P + \mathbf{B}^{(k)}) + \mathbf{I}_M^{(k)} \quad (2)$$

$$\begin{aligned} \frac{\partial(r^{(k)}\rho^{(k)}T^{(k)})}{\partial t} + \nabla \cdot (r^{(k)}\rho^{(k)}\mathbf{u}^{(k)}T^{(k)}) = \nabla \cdot \left[ r^{(k)} \left( \frac{\mu^{(k)}}{Pr^{(k)}} + \frac{\mu_t^{(k)}}{Pr_t^{(k)}} \right) \nabla T^{(k)} \right] \\ + \frac{r^{(k)}}{c_p^{(k)}} \left\{ \beta^{(k)}T^{(k)} \left[ \frac{\partial P}{\partial t} + \nabla \cdot (P\mathbf{u}^{(k)}) - P\nabla \cdot (\mathbf{u}^{(k)}) \right] + \Phi^{(k)} + \dot{q}^{(k)} \right\} + \frac{I_E^{(k)}}{c_p^{(k)}} \end{aligned} \quad (3)$$

where the meanings of the various terms are as given in the nomenclature.

The effect of turbulence on interfacial mass, momentum, and energy transfer is difficult to model and is still an active area of research. Similar to single-fluid flow, researchers have advertised several flow-dependent models to describe turbulence. These models vary in complexity from simple algebraic [47] models to state-of-the-art Reynolds-stress [48]

models. However, the one used here is the two-equation  $k$ - $\epsilon$  model [49] described next. The phasic conservation equations governing the turbulence kinetic energy ( $k$ ) and turbulence dissipation rate ( $\epsilon$ ) for the  $k^{\text{th}}$  fluid are given by:

$$\frac{\partial(\mathbf{r}^{(k)} \rho^{(k)} k^{(k)})}{\partial t} + \nabla \cdot (\mathbf{r}^{(k)} \rho^{(k)} \mathbf{u}^{(k)} k^{(k)}) = \nabla \cdot \left( \mathbf{r}^{(k)} \frac{\mu_t^{(k)}}{\sigma_k^{(k)}} \nabla k^{(k)} \right) + \mathbf{r}^{(k)} \rho^{(k)} (G^{(k)} - \epsilon^{(k)}) + I_k^{(k)} \quad (4)$$

$$\begin{aligned} \frac{\partial(\mathbf{r}^{(k)} \rho^{(k)} \epsilon^{(k)})}{\partial t} + \nabla \cdot (\mathbf{r}^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \epsilon^{(k)}) = & \nabla \cdot \left( \mathbf{r}^{(k)} \frac{\mu_t^{(k)}}{\sigma_\epsilon^{(k)}} \nabla \epsilon^{(k)} \right) + \\ & \mathbf{r}^{(k)} \rho^{(k)} \frac{\epsilon^{(k)}}{k^{(k)}} (c_{1\epsilon} G^{(k)} - c_{2\epsilon} \epsilon^{(k)}) + I_\epsilon^{(k)} \end{aligned} \quad (5)$$

where  $I_k^{(k)}$  and  $I_\epsilon^{(k)}$  represent the interfacial turbulence terms. The turbulent viscosity is calculated as:

$$\mu_t^{(k)} = C_\mu \frac{[k^{(k)}]^2}{\epsilon^{(k)}} \quad (6)$$

For two-phase flow, several extensions of the  $k$ - $\epsilon$  model that are based on calculating the turbulent viscosity by solving the  $k$  and  $\epsilon$  equations for the carrier or continuous phase only have been proposed in the literature [50-55]. In a recent article, Cokljat and Ivanov [49] presented a phase coupled  $k$ - $\epsilon$  turbulence model, intended for the cases where a non-dilute secondary phase is present, in which the  $k$ - $\epsilon$  transport equations for all phases are solved. Since the method is still not well developed, the first approach in which only the  $k$  and  $\epsilon$  equations for the carrier phase are solved is adopted in this work. Details regarding the specific model used will be presented as needed.

If a typical representative variable associated with phase ( $k$ ) is denoted by  $\phi^{(k)}$ , equations (1)-(5) can be presented via the following general phasic equation:

$$\frac{\partial(\mathbf{r}^{(k)} \rho^{(k)} \phi^{(k)})}{\partial t} + \nabla \cdot (\mathbf{r}^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \phi^{(k)}) = \nabla \cdot (\mathbf{r}^{(k)} \Gamma^{(k)} \nabla \phi^{(k)}) + \mathbf{r}^{(k)} Q^{(k)} \quad (7)$$

where the expression for  $\Gamma^{(k)}$  and  $Q^{(k)}$  can be deduced from the parent equations.

The above set of differential equations has to be solved in conjunction with constraints on certain variables represented by algebraic relations. These auxiliary relations include the equations of state, the geometric conservation equation, and the interfacial mass, momentum, energy, and turbulence energy transfers.

Physically, the geometric conservation equation is a statement indicating that the sum of volumes occupied by the different fluids within a cell is equal to the volume of the cell containing the fluids.

$$\sum_k r^{(k)} = 1 \quad (8)$$

For a compressible multi-phase flow, auxiliary equations of state relating density to pressure and temperature are needed. For the  $k^{\text{th}}$  phase, such an equation can be written as:

$$\rho^{(k)} = \rho^{(k)}(P, T^{(k)}) \quad (9)$$

Several models have been developed for computing the interfacial mass, momentum, energy, and turbulence energy transfers terms. The closures used in this work will be detailed whenever they arise while solving problems.

In order to present a complete mathematical problem, thermodynamic relations may be needed and initial and boundary conditions should supplement the above equations.

## Discretization Procedure

The general conservation equation (7) is integrated over a finite volume to yield:

$$\begin{aligned} \iint_{\Omega} \frac{\partial(r^{(k)} \rho^{(k)} \phi^{(k)})}{\partial t} d\Omega + \iint_{\Omega} \nabla \cdot (r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \phi^{(k)}) d\Omega \\ = \iint_{\Omega} \nabla \cdot (r^{(k)} \Gamma^{(k)} \nabla \phi^{(k)}) d\Omega + \iint_{\Omega} r^{(k)} Q^{(k)} d\Omega \end{aligned} \quad (10)$$

Where  $\Omega$  is the volume of the control cell (Fig. 1(a)). Using the divergence theorem to transform the volume integral into a surface integral and then replacing the surface integral by a summation of the fluxes over the sides of the control volume, equation (10) is transformed to:

$$\frac{\partial(r^{(k)}\rho^{(k)}\phi^{(k)}\Omega)}{\partial t} + \sum_{nb=e,w,n,s,t,b} (\mathbf{J}_{nb}^{(k)D} + \mathbf{J}_{nb}^{(k)C}) = r^{(k)}Q^{(k)}\Omega \quad (11)$$

where  $\mathbf{J}_{nb}^{(k)D}$  and  $\mathbf{J}_{nb}^{(k)C}$  are the diffusive and convective fluxes, respectively. The discretization of the diffusion term is second order accurate and follows the derivations presented in [35]. For the convective terms, the High Resolution SMART [13] scheme is employed, even for the calculation of interface densities, and applied within the context of the NVSF methodology [15]. Substituting the face values by their functional relationship relating to the node values of  $\phi$ , Eq. (11) is transformed after some algebraic manipulations into the following discretized equation:

$$A_P^{(k)}\phi_P^{(k)} = \sum_{NB} A_{NB}^{(k)}\phi_{NB}^{(k)} + B_P^{(k)} \quad (12)$$

where the coefficients  $A_P^{(k)}$  and  $A_{NB}^{(k)}$  depend on the selected scheme and  $B_P^{(k)}$  is the source term of the discretized equation. In compact form, the above equation can be written as

$$\phi_P^{(k)} = H_P[\phi^{(k)}] = \frac{\sum_{NB} A_{NB}^{(k)}\phi_{NB}^{(k)} + B_P^{(k)}}{A_P^{(k)}} \quad (13)$$

The discretization procedure for the momentum equation yields an algebraic equation of the form:

$$\mathbf{u}_P^{(k)} = \mathbf{H}_P[\mathbf{u}^{(k)}] - r^{(k)}\mathbf{D}_P^{(k)}\nabla_P(P) \quad (14)$$

On the other hand, the phasic mass-conservation equation (Eq. (1)) can be either viewed as a phasic volume fraction equation:

$$r_P^{(k)} = H_P[r^{(k)}] \quad (15)$$

or as the following phasic continuity equation to be used in deriving the pressure correction equation:

$$\frac{(r_P^{(k)}\rho_P^{(k)}) - (r_P^{(k)}\rho_P^{(k)})^{Old}}{\delta t}\Omega + \Delta_P[r^{(k)}\rho^{(k)}\mathbf{u}^{(k)} \cdot \mathbf{S}] = r^{(k)}\dot{M}^{(k)} \quad (16)$$

where the  $\Delta$  operator represents the following operation:

$$\Delta_P[\Theta] = \sum_{f=NB(P)} \Theta_f \quad (17)$$

## Geometric Conservation Based Algorithms (GCBA)

The numbers of equations describing an  $n$ -fluid flow situation are:  $n$  momentum equations,  $n$  volume fraction (or mass conservation) equations, a geometric conservation equation, and for the case of a compressible flow an additional  $n$  auxiliary pressure-density relations. Moreover, the variables involved are the  $n$  velocity vectors, the  $n$  volume fractions, the pressure field, and for a compressible flow an additional  $n$  unknown density fields. It is clear that the  $n$ -velocity fields are associated with the  $n$ -momentum equations, i.e. the momentum equations can be used directly to calculate the velocity fields. The volume fractions could arguably be calculated from the volume fraction equations, which means that the remaining equation i.e. the geometric conservation equation (the volume fractions sum to 1) has to be used in deriving the pressure equation, or equivalently the pressure correction equation. This results in what is called here the **Geometric Conservation Based Algorithm (GCBA)**.

The sequence of events in the Geometric Conservation Based Algorithm (GCBA) is as follows:

- Solve the individual mass conservation equations for volume fractions.
- Solve the momentum equations for velocities.
- Solve the pressure correction equation.
- Correct velocity, volume fraction, density, and pressure fields.
- Solve the individual energy equations.
- Return to the first step and repeat until convergence.

The GCBA uses the momentum equations for a first estimate of velocities. However, the volume fractions are calculated without enforcing the geometric conservation equation. Hence, the mass conservation equations of all fluids are used to calculate the volume fractions. As such, the pressure correction equation should be based on the geometric conservation equation and used to restore the imbalance of volume fractions. The errors in

the calculated volume fractions are expressed in terms of pressure correction ( $P'$ ), which is also used to adjust the velocity and density fields.

## The Pressure Correction Equation

After solving the continuity equations for the volume fraction fields and the momentum equations for the velocity fields, the next step is to correct the various fields such that the volume fraction fields satisfy the compatibility equation and the velocity and pressure fields satisfy the continuity equations. For that purpose, a guess-and-correct scheme is adopted. Correction is obtained by solving a pressure correction equation derived from the geometric conservation equation. To start the derivation, it is noticed that initially the volume fraction fields denoted by  $r^{(k)*}$ , do not satisfy the compatibility equation and a discrepancy exists i.e.

$$\text{RESG}_p = 1 - \sum_k r_p^{(k)*} \quad (18)$$

A change to  $r^{(k)*}$  is sought that would restore the balance. The corrected  $r$  value, denoted by  $r^{(k)}$  ( $r^{(k)} = r^{(k)*} + r^{(k)'}$ ), is such that

$$\sum_k (r^{(k)'}) = 1 - \sum_k (r^{(k)*}) = \text{RESG}_p \quad (19)$$

Correction to the volume fraction,  $r^{(k)'}$ , will be associated with a correction to the velocity, density, and pressure fields,  $\mathbf{u}^{(k)'}$ ,  $\rho^{(k)'}$ , and  $P'$  respectively. Thus, the corrected fields are given as:

$$r^{(k)} = r^{(k)*} + r^{(k)'}, P = P^\circ + P', \mathbf{u}^{(k)} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, \rho^{(k)} = \rho^{(k)\circ} + \rho^{(k)'} \quad (20)$$

The discretized form of the corrected continuity equation of phase (k) can be written as

$$\frac{(r_p^{(k)*} + r_p^{(k)'}) (\rho_p^{(k)\circ} + \rho_p^{(k)'}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left( (r^{(k)*} + r^{(k)'}) (\rho^{(k)\circ} + \rho^{(k)'}) (\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}) \cdot \mathbf{S} \right) = \dot{M}_p^{(k)} (r_p^{(k)*} + r_p^{(k)'}) \Omega_p \quad (21)$$

Neglecting second and third order terms (i.e.  $r_p^{(k)'} \rho_p^{(k)'}$ ,  $\rho_p^{(k)'} \mathbf{u}^{(k)'}$ ,  $r_p^{(k)'} \mathbf{u}^{(k)'}$ , and  $r_p^{(k)'} \rho_p^{(k)'} \mathbf{u}^{(k)'}$ ), its

expanded form reduces to:

$$\begin{aligned}
& \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)'} + \mathbf{r}_p^{(k)'} \rho_p^{(k)\circ})}{\delta t} \Omega_p + \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{u}^{(k)'} \cdot \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} \rho_p^{(k)'} + \rho^{(k)\circ} \mathbf{U}^{(k)*} \mathbf{r}^{(k)'}) \right] \\
& - \dot{\mathbf{M}}_p^{(k)} \mathbf{r}_p^{(k)'} \Omega_p = - \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)\circ}) - (\mathbf{r}_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p - \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*}) \right] + \dot{\mathbf{M}}_p^{(k)} \mathbf{r}_p^{(k)*} \Omega_p
\end{aligned} \tag{22}$$

Writing  $\mathbf{u}^{(k)'}$  as a function of  $P'$ , similar to what is usually done in a SIMPLE-like algorithm, the correction momentum equations become

$$\mathbf{u}^{(k)'} = \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla P' - \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla P^\circ - \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla P' \tag{23}$$

Substituting Eq. (23) into Eq. (22), rearranging, and discretizing one gets

$$\begin{aligned}
& \mathbf{r}_p^{(k)'} - \mathbf{H}_p \left[ \mathbf{r}^{(k)'} \right] = \\
& - \mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla P' \right) \cdot \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} \rho_p^{(k)'} \right] \right. \\
& \left. + \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)\circ}) - (\mathbf{r}_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*}) \right] - \dot{\mathbf{M}}_p^{(k)} \mathbf{r}_p^{(k)*} \Omega_p \right)
\end{aligned} \tag{24}$$

where  $\mathbf{R}_p^{(k)} = 1 / \mathbf{A}_p^{(k)}$ .

Neglecting the correction to neighboring cells, equation (24) reduces to:

$$\begin{aligned}
& \mathbf{r}_p^{(k)'} = - \mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla P' \right) \cdot \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} \rho_p^{(k)'} \right] \right. \\
& \left. + \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)\circ}) - (\mathbf{r}_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*}) \right] - \dot{\mathbf{M}}_p^{(k)} \mathbf{r}_p^{(k)*} \Omega_p \right)
\end{aligned} \tag{25}$$

Substituting this equation into the geometric conservation equation and replacing density

correction in terms of pressure correction (i.e.  $\rho^{(k)'} = C_\rho^{(k)} P'$ ), the pressure correction equation

is obtained as

$$\sum_k \left\{ - \mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p C_\rho^{(k)}}{\delta t} P' + \Delta_p \left[ \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} C_\rho^{(k)} P' \right] + \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla P' - \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla P' \right) \cdot \mathbf{S} \right] \right. \right. \\
\left. \left. + \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)\circ}) - (\mathbf{r}_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*}) \right] \right) \right\} = \text{RESG}_p \tag{26}$$

If the  $\mathbf{HP}[\mathbf{u}^{(k)'}]$  term in the above equation is retained, there will result a pressure correction equation relating the pressure correction value at a point to all values in the domain. To facilitate implementation and reduce cost, simplifying assumptions related to this term have been introduced. Depending on these assumptions, different algorithms are obtained. A summary of the various GCBA algorithms (GCBA-SIMPLE, GCBA-SIMPLEC, GCBA-PISO,...) used in this work is given next.

### ***The GCBA following SIMPLE (GCBA-SIMPLE): Symbolic Form***

Predictor:

$$\mathbf{r}_p^{(k)*} = \mathbf{H}_p[\mathbf{r}^{(k)*}] \quad (27)$$

$$\mathbf{u}_p^{(k)*} = \mathbf{HP}_p[\mathbf{u}^{(k)*}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^\circ \quad (28)$$

Corrector:

$$\left( \mathbf{u}^{(k)'}, P', \rho^{(k)'}, \mathbf{r}^{(k)'} \left\{ \begin{array}{l} \mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \\ \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}, \mathbf{r}^{(k)**} = \mathbf{r}^{(k)*} + \mathbf{r}^{(k)'} \end{array} \right. \right) \quad (29)$$

$$\therefore \mathbf{u}_p^{(k)**} = \mathbf{HP}_p[\mathbf{u}^{(k)**}] - \mathbf{r}_p^{(k)**} \mathbf{D}_p^{(k)} \nabla_p P^* = \mathbf{HP}_p[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - (\mathbf{r}_p^{(k)*} + \mathbf{r}_p^{(k)'}) \mathbf{D}_p^{(k)} \nabla_p (P^\circ + P') \quad (30)$$

$$\therefore \left\{ \begin{array}{l} \mathbf{u}_p^{(k)'} = \mathbf{HP}_p[\mathbf{u}^{(k)'}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P^\circ - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \\ \mathbf{r}_p^{(k)'} = -\mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{u}^{(k)'}) \cdot \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} \rho^{(k)'} \right] \right) \end{array} \right. \quad (31)$$

Condition:

$$\sum_k \{\mathbf{r}_p^{(k)'}\} = \text{RESG}_p \quad (32)$$

$$\therefore \sum_k \left\{ -\mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p C_\rho^{(k)}}{\delta t} P' + \Delta_p [\mathbf{r}^{(k)*} \mathbf{U}^{(k)*} C_\rho^{(k)} P'] + \Delta_p [\mathbf{r}^{(k)*} \rho^{(k)\circ} (\mathbf{HP}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla P' - \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] + \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)\circ}) - (\mathbf{r}_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*})] \right) \right\} = \text{RESG}_p \quad (33)$$

$$\sum_k \left\{ R_p^{(k)} \left( \frac{r_p^{(k)*} \Omega_p C_\rho^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_\rho^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right) \right\} =$$

$$\therefore - \sum_k \left\{ R_p^{(k)} \left( \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{Old}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] + \right. \right. \\ \left. \left. \Delta_p [r^{(k)*} \rho^{(k)\circ} (\mathbf{H}P[\mathbf{u}^{(k)'}] - r^{(k)'} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right) \right\} - \text{RESG}_p \quad (34)$$

Approximation:

Neglect:  $\mathbf{H}P[\mathbf{u}^{(k)'}]$ ,  $r^{(k)'} \mathbf{D}^{(k)} \nabla P'$

$$\Rightarrow \mathbf{u}_p^{(k)'} = -r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' \quad (35)$$

Approximate Equation:

$$\sum_k \left\{ R_p^{(k)} \left( \frac{r_p^{(k)*} \Omega_p C_\rho^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_\rho^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right) \right\} =$$

$$\Rightarrow - \sum_k \left\{ R_p^{(k)} \left( \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{Old}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] \right) \right\} - \text{RESG}_p \quad (36)$$

## A Global GCBA-SIMPLE Iteration

- 
- Solve implicitly for the volume fraction fields.
  - Solve implicitly for  $\mathbf{u}^{(k)}$ , using the old pressure, density, and volume fraction fields.
  - Calculate the  $\mathbf{D}^{(k)}$  fields.
  - Solve the pressure correction equation.
  - Correct  $\mathbf{u}^{(k)}$ ,  $P$ ,  $r^{(k)}$  and  $\rho^{(k)}$ .
  - Solve implicitly the energy equations and update the density fields.
  - Return to the first step and iterate until convergence.
- 

## The GCBA following SIMPLEC (GCBA-SIMPLEC): Symbolic Form

Predictor:

$$r_p^{(k)*} = H_p[r^{(k)*}] \quad (37)$$

$$\mathbf{u}_p^{(k)*} = \mathbf{H}P_p[\mathbf{u}^{(k)*}] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^\circ \quad (38)$$

Corrector:

$$\left( \mathbf{u}^{(k)'}, P', \rho^{(k)'}, r^{(k)'} \left\{ \begin{aligned} \mathbf{u}^{(k)**} &= \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'} \\ P^* &= P^\circ + P' \\ \rho^{(k)*} &= \rho^{(k)\circ} + \rho^{(k)'} \\ r^{(k)**} &= r^{(k)*} + r^{(k)'} \end{aligned} \right. \right) \quad (39)$$

$$\therefore \mathbf{u}_p^{(k)**} = \mathbf{H}P_p[\mathbf{u}^{(k)**}] - r_p^{(k)**} \mathbf{D}_p^{(k)} \nabla_p P^* = \mathbf{H}P_p[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - (r_p^{(k)*} + r_p^{(k)'}) \mathbf{D}_p^{(k)} \nabla_p (P^\circ + P') \quad (40)$$

$$\therefore \mathbf{u}_p^{(k)'} = \mathbf{H}P_p[\mathbf{u}^{(k)'}] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' - r_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P^\circ - r_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P' \quad (41)$$

Subtracting  $\tilde{\mathbf{H}P}_p[\mathbf{I}]\mathbf{u}_p^{(k)'}$  from both sides, one gets

$$\mathbf{u}_p^{(k)'} - \tilde{\mathbf{H}}\mathbf{P}_p[1]\mathbf{u}_p^{(k)'} = \begin{pmatrix} \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'}] - \tilde{\mathbf{H}}\mathbf{P}_p[1]\mathbf{u}_p^{(k)'} - \mathbf{r}_p^{(k)*}\mathbf{D}_p^{(k)}\nabla_p\mathbf{P}' \\ -\mathbf{r}_p^{(k)'}\mathbf{D}_p^{(k)}\nabla_p\mathbf{P}^\circ - \mathbf{r}_p^{(k)'}\mathbf{D}_p^{(k)}\nabla_p\mathbf{P}' \end{pmatrix} \quad (42)$$

$$\therefore \left(1 - \tilde{\mathbf{H}}\mathbf{P}_p[1]\right)\mathbf{u}_p^{(k)'} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'} - \mathbf{u}_p^{(k)'}] - \mathbf{r}_p^{(k)*}\mathbf{D}_p^{(k)}\nabla_p\mathbf{P}' - \mathbf{r}_p^{(k)'}\mathbf{D}_p^{(k)}\nabla_p\mathbf{P}^\circ - \mathbf{r}_p^{(k)'}\mathbf{D}_p^{(k)}\nabla_p\mathbf{P}' \quad (43)$$

$$\therefore \mathbf{u}_p^{(k)'} = \begin{bmatrix} \frac{\mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'} - \mathbf{u}_p^{(k)'}]}{\left(1 - \tilde{\mathbf{H}}\mathbf{P}_p[1]\right)} - \mathbf{r}_p^{(k)*}\frac{\mathbf{D}_p^{(k)}}{\left(1 - \tilde{\mathbf{H}}\mathbf{P}_p[1]\right)}\nabla_p\mathbf{P}' \\ -\mathbf{r}_p^{(k)'}\frac{\mathbf{D}_p^{(k)}}{\left(1 - \tilde{\mathbf{H}}\mathbf{P}_p[1]\right)}\nabla_p\mathbf{P}^\circ - \mathbf{r}_p^{(k)'}\frac{\mathbf{D}_p^{(k)}}{\left(1 - \tilde{\mathbf{H}}\mathbf{P}_p[1]\right)}\nabla_p\mathbf{P}' \end{bmatrix} \quad (44)$$

$$\therefore \begin{cases} \mathbf{u}_p^{(k)'} = \frac{\mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'} - \mathbf{u}_p^{(k)'}]}{\left(1 - \tilde{\mathbf{H}}\mathbf{P}_p[1]\right)} - \mathbf{r}_p^{(k)*}\tilde{\mathbf{D}}_p^{(k)}\nabla_p\mathbf{P}' - \mathbf{r}_p^{(k)'}\tilde{\mathbf{D}}_p^{(k)}\nabla_p\mathbf{P}^\circ - \mathbf{r}_p^{(k)'}\tilde{\mathbf{D}}_p^{(k)}\nabla_p\mathbf{P}' \\ \rho^{(k)'} = C_\rho^{(k)}\mathbf{P}' \\ \mathbf{r}_p^{(k)'} = -\mathbf{R}_p^{(k)}\left(\frac{\mathbf{r}_p^{(k)*}\Omega_p}{\delta t}\rho_p^{(k)'} + \Delta_p\left[\left(\mathbf{r}^{(k)*}\rho^{(k)\circ}\mathbf{u}^{(k)'}\right).\mathbf{S} + \mathbf{r}^{(k)*}\mathbf{U}^{(k)*}\rho^{(k)'}\right]\right) \end{cases} \quad (45)$$

Condition:

$$\sum_k \{\mathbf{r}_p^{(k)'}\} = \text{RESG}_p \quad (46)$$

$$\therefore \sum_k \left\{ -\mathbf{R}_p^{(k)} \left[ \Delta_p \left[ \mathbf{r}^{(k)*}\rho^{(k)\circ} \left( \frac{\mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'} - \mathbf{u}_p^{(k)'}]}{\left(1 - \tilde{\mathbf{H}}\mathbf{P}_p[1]\right)} - \mathbf{r}^{(k)*}\tilde{\mathbf{D}}_p^{(k)}\nabla_p\mathbf{P}' - \mathbf{r}^{(k)'}\tilde{\mathbf{D}}_p^{(k)}\nabla_p\mathbf{P}' \right) \right] \cdot \mathbf{S} \right] \right. \\ \left. + \frac{\left(\mathbf{r}_p^{(k)*}\rho_p^{(k)\circ}\right) - \left(\mathbf{r}_p^{(k)}\rho_p^{(k)}\right)^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ \left(\mathbf{r}^{(k)*}\rho^{(k)\circ}\mathbf{U}^{(k)*}\right) \right] \right\} = \text{RESG}_p \quad (47)$$

$$\sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*}\Omega_p C_\rho^{(k)}}{\delta t} \mathbf{P}' + \Delta_p \left[ \mathbf{r}^{(k)*}\mathbf{U}^{(k)*} C_\rho^{(k)} \mathbf{P}' \right] - \Delta_p \left[ \mathbf{r}^{(k)*}\rho^{(k)\circ} \left( \mathbf{r}^{(k)*}\tilde{\mathbf{D}}_p^{(k)}\nabla_p\mathbf{P}' \right) \cdot \mathbf{S} \right] \right) \right\} = \\ - \sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{\left(\mathbf{r}_p^{(k)*}\rho_p^{(k)\circ}\right) - \left(\mathbf{r}_p^{(k)}\rho_p^{(k)}\right)^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ \left(\mathbf{r}^{(k)*}\rho^{(k)\circ}\mathbf{U}^{(k)*}\right) \right] + \right. \\ \left. \Delta_p \left[ \mathbf{r}^{(k)*}\rho^{(k)\circ} \left( \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'} - \mathbf{u}_p^{(k)'}] - \mathbf{r}^{(k)'}\tilde{\mathbf{D}}_p^{(k)}\nabla_p\mathbf{P}' \right) \cdot \mathbf{S} \right] \right) \right\} - \text{RESG}_p \quad (48)$$

Approximation:

Neglect:  $\mathbf{HP}[\mathbf{u}^{(k)'} - \mathbf{u}_p^{(k)'}], r^{(k)'} \tilde{\mathbf{D}}^{(k)} \nabla P'$

$$\Rightarrow \mathbf{u}_p^{(k)'} = -\mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' \quad (49)$$

Approximate Equation:

$$\begin{aligned} \Rightarrow \sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P' + \Delta_p [\mathbf{r}^{(k)*} \mathbf{U}^{(k)*} C_p^{(k)} P'] - \Delta_p [\mathbf{r}^{(k)*} \rho^{(k)\circ} (\mathbf{r}^{(k)*} \tilde{\mathbf{D}}^{(k)} \nabla P', \mathbf{S})] \right) \right\} = \\ - \sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)\circ}) - (\mathbf{r}_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*}] \right) \right\} - \text{RESG}_p \end{aligned} \quad (50)$$

**A Global GCBA-SIMPLEC Iteration**

- 
- Solve implicitly for the volume fraction fields.
  - Solve implicitly for  $\mathbf{u}^{(k)}$ , using the old pressure, density, and volume fraction fields.
  - Calculate the  $\tilde{\mathbf{D}}^{(k)}$  fields.
  - Solve the pressure correction equation.
  - Correct  $\mathbf{u}^{(k)}$ ,  $P$ ,  $r^{(k)}$  and  $\rho^{(k)}$ .
  - Solve implicitly the energy equations and update the density fields.
  - Return to the first step and iterate until convergence.
- 

**The GCBA following PRIME (GCBA-PRIME): Symbolic Form**Predictor:

$$\mathbf{r}_p^{(k)*} = \mathbf{H}_p[\mathbf{r}^{(k)\circ}] \quad (51)$$

$$\mathbf{u}_p^{(k)*} = \mathbf{HP}_p[\mathbf{u}^{(k)\circ}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^\circ \quad (52)$$

Corrector:

$$\left( \mathbf{u}^{(k)'}, P', \rho^{(k)'}, \mathbf{r}^{(k)'} \left\{ \begin{aligned} \mathbf{u}^{(k)**} &= \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \\ \rho^{(k)*} &= \rho^{(k)\circ} + \rho^{(k)'}, \mathbf{r}^{(k)**} = \mathbf{r}^{(k)*} + \mathbf{r}^{(k)'} \end{aligned} \right. \right) \quad (53)$$

$$\therefore \mathbf{u}_p^{(k)**} = \mathbf{HP}_p[\mathbf{u}^{(k)**}] - \mathbf{r}_p^{(k)**} \mathbf{D}_p^{(k)} \nabla_p P^* = \mathbf{HP}_p[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - (\mathbf{r}_p^{(k)*} + \mathbf{r}_p^{(k)'}) \mathbf{D}_p^{(k)} \nabla_p (P^\circ + P') \quad (54)$$

$$\therefore \left\{ \begin{aligned} \mathbf{u}_p^{(k)'} &= \mathbf{HP}_p[\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] + \mathbf{HP}_p[\mathbf{u}^{(k)'}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P^\circ - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P' \\ \rho^{(k)'} &= C_p^{(k)} P' \\ \mathbf{r}_p^{(k)'} &= -\mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p [(\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{u}^{(k)'}) \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} \rho^{(k)'}] \right) \end{aligned} \right. \quad (55)$$

Condition:

$$\sum_k \{\mathbf{r}_p^{(k)'}\} = \text{RESG}_p \quad (56)$$

$$\therefore \sum_k \left\{ -R_p^{(k)} \left[ \frac{r_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_p^{(k)} P'] + \Delta_p \left[ r^{(k)*} \rho^{(k)\circ} \left( \mathbf{HP}[\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] + \mathbf{HP}[\mathbf{u}^{(k)'}] \right) \cdot \mathbf{S} \right] + \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] \right] \right\} = \text{RESG}_p \quad (57)$$

$$\sum_k \left\{ R_p^{(k)} \left[ \frac{r_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_p^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right] \right\} =$$

$$\therefore - \sum_k \left\{ R_p^{(k)} \left[ \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] + \Delta_p [r^{(k)*} \rho^{(k)\circ} (\mathbf{HP}[\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] + \mathbf{HP}[\mathbf{u}^{(k)'}] - r^{(k)'} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right] \right\} - \text{RESG}_p \quad (58)$$

Approximation:

Neglect:  $\mathbf{HP}[\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}]$ ,  $\mathbf{HP}[\mathbf{u}^{(k)'}]$ ,  $r^{(k)'} \mathbf{D}^{(k)} \nabla P'$

$$\Rightarrow \mathbf{u}_p^{(k)'} = -r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' \quad (59)$$

Approximate Equation:

$$\sum_k \left\{ R_p^{(k)} \left[ \frac{r_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_p^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right] \right\} =$$

$$\Rightarrow - \sum_k \left\{ R_p^{(k)} \left[ \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] \right] \right\} - \text{RESG}_p \quad (60)$$

## A Global GCBA-PRIME Iteration

- 
- Solve explicitly for the volume fraction fields.
  - Solve explicitly for  $\mathbf{u}^{(k)}$ , using the old pressure and density fields.
  - Calculate the  $\mathbf{D}^{(k)}$  fields.
  - Solve the pressure correction equation.
  - Correct  $\mathbf{u}^{(k)}$ ,  $P$ ,  $r^{(k)}$  and  $\rho^{(k)}$ .
  - Solve implicitly the energy equations and update the density fields.
  - Return to the first step and iterate until convergence.
- 

## The GCBA following SIMPLEST (GCBA-SIMPLEST): Symbolic Form

Predictor:

$$r_p^{(k)*} = H_p^D [r^{(k)*}] + H_p^C [r^{(k)\circ}] \quad (61)$$

$$\mathbf{u}_p^{(k)*} = \mathbf{HP}_p^D [\mathbf{u}^{(k)*}] + \mathbf{HP}_p^C [\mathbf{u}^{(k)\circ}] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^\circ \quad (62)$$

Corrector:

$$\left( \mathbf{u}^{(k)'} , P' , \rho^{(k)'} , \mathbf{r}^{(k)'} \left\{ \begin{array}{l} \mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'} , P^* = P^\circ + P' , \\ \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'} , \mathbf{r}^{(k)**} = \mathbf{r}^{(k)*} + \mathbf{r}^{(k)'} \end{array} \right. \right) \quad (63)$$

$$\begin{aligned} \mathbf{u}_P^{(k)**} &= \mathbf{HP}_P^D [\mathbf{u}^{(k)**}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)**}] - \mathbf{r}_P^{(k)**} \mathbf{D}_P^{(k)} \nabla_P P^* \\ &= \mathbf{HP}_P^D [\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - (\mathbf{r}_P^{(k)*} + \mathbf{r}_P^{(k)'}) \mathbf{D}_P^{(k)} \nabla_P (P^\circ + P') \\ \therefore &= \mathbf{HP}_P^D [\mathbf{u}^{(k)*}] + \mathbf{HP}_P^D [\mathbf{u}^{(k)'}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)*}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)'}] \\ &\quad - \mathbf{r}_P^{(k)*} \mathbf{D}_P^{(k)} \nabla_P P^\circ - \mathbf{r}_P^{(k)*} \mathbf{D}_P^{(k)} \nabla_P P' - \mathbf{r}_P^{(k)'} \mathbf{D}_P^{(k)} \nabla_P P^\circ - \mathbf{r}_P^{(k)'} \mathbf{D}_P^{(k)} \nabla_P P' \end{aligned} \quad (64)$$

$$\therefore \left\{ \begin{array}{l} \mathbf{u}_P^{(k)'} = \mathbf{HP}_P [\mathbf{u}^{(k)'}] + \mathbf{HP}_P^C [\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] - \mathbf{r}_P^{(k)*} \mathbf{D}_P^{(k)} \nabla_P P' - \mathbf{r}_P^{(k)'} \mathbf{D}_P^{(k)} \nabla_P P^\circ - \mathbf{r}_P^{(k)'} \mathbf{D}_P^{(k)} \nabla_P P' \\ \rho^{(k)'} = C_\rho^{(k)} P' \\ \mathbf{r}_P^{(k)'} = -\mathbf{R}_P^{(k)} \left( \frac{\mathbf{r}_P^{(k)*} \Omega_P}{\delta t} \rho_P^{(k)'} + \Delta_P \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{u}^{(k)'}), \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} \rho^{(k)'} \right] \right) \end{array} \right. \quad (65)$$

Condition:

$$\sum_k \{ \mathbf{r}_P^{(k)'} \} = \text{RESG}_P \quad (66)$$

$$\therefore \sum_k \left\{ -\mathbf{R}_P^{(k)} \left[ \begin{array}{l} \frac{\mathbf{r}_P^{(k)*} \Omega_P C_\rho^{(k)}}{\delta t} P' + \Delta_P [\mathbf{r}^{(k)*} \mathbf{U}^{(k)*} C_\rho^{(k)} P'] + \\ \Delta_P \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{HP}[\mathbf{u}^{(k)'}] + \mathbf{HP}^C [\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] \right) \mathbf{S} \right] \\ + \frac{(\mathbf{r}_P^{(k)*} \rho_P^{(k)\circ}) - (\mathbf{r}_P^{(k)} \rho_P^{(k)})^{\text{Old}}}{\delta t} \Omega_P + \Delta_P [(\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*})] \end{array} \right] \right\} = \text{RESG}_P \quad (67)$$

$$\begin{aligned} &\sum_k \left\{ \mathbf{R}_P^{(k)} \left( \frac{\mathbf{r}_P^{(k)*} \Omega_P C_\rho^{(k)}}{\delta t} P' + \Delta_P [\mathbf{r}^{(k)*} \mathbf{U}^{(k)*} C_\rho^{(k)} P'] - \Delta_P [\mathbf{r}^{(k)*} \rho^{(k)\circ} (\mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla_P P') \cdot \mathbf{S}] \right) \right\} = \\ \therefore &-\sum_k \left\{ \mathbf{R}_P^{(k)} \left( \frac{(\mathbf{r}_P^{(k)*} \rho_P^{(k)\circ}) - (\mathbf{r}_P^{(k)} \rho_P^{(k)})^{\text{Old}}}{\delta t} \Omega_P + \Delta_P [(\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*})] + \right. \right. \\ &\quad \left. \left. \Delta_P [\mathbf{r}^{(k)*} \rho^{(k)\circ} (\mathbf{HP}^C [\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}] + \mathbf{HP}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla_P P') \cdot \mathbf{S}] \right) \right\} - \text{RESG}_P \end{aligned} \quad (68)$$

Approximation:

Neglect:  $\mathbf{HP}^C [\mathbf{u}^{(k)*} - \mathbf{u}^{(k)\circ}]$ ,  $\mathbf{HP}[\mathbf{u}^{(k)'}]$ ,  $\mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla_P P'$

$$\Rightarrow \mathbf{u}_P^{(k)'} = -\mathbf{r}_P^{(k)*} \mathbf{D}_P^{(k)} \nabla_P P' \quad (69)$$

Approximate Equation:

$$\begin{aligned} \Rightarrow \sum_k \left\{ R_p^{(k)} \left( \frac{r_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_p^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right) \right\} = \\ - \sum_k \left\{ R_p^{(k)} \left( \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] \right) \right\} - \text{RESG}_p \end{aligned} \quad (70)$$

### A Global GCBA-SIMPLEST Iteration

- 
- Solve for  $r^{(k)}$ , treating  $H^D[r^{(k)}]$  implicitly and  $H^C[r^{(k)}]$  explicitly.
  - Solve for  $u^{(k)}$ , treating  $HP^D[u^{(k)}]$  implicitly and  $HP^C[u^{(k)}]$  explicitly.
  - Calculate the  $\mathbf{D}^{(k)}$  fields.
  - Solve the pressure correction equation.
  - Correct  $u^{(k)}$ ,  $P$ ,  $r^{(k)}$  and  $\rho^{(k)}$ .
  - Solve implicitly the energy equations and update the density fields.
  - Return to the first step and iterate until convergence.
- 

### The GCBA following PISO (GCBA-PISO): Symbolic Form

First Predictor:

$$r_p^{(k)*} = H_p[r^{(k)*}] \quad (71)$$

$$u_p^{(k)*} = HP_p[u^{(k)*}] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^\circ \quad (72)$$

First Corrector:

$$(u^{(k)'}, P', \rho^{(k)'}, r^{(k)'}) \left( \begin{aligned} u^{(k)**} &= u^{(k)*} + u^{(k)'}, P^* = P^\circ + P', \\ \rho^{(k)*} &= \rho^{(k)\circ} + \rho^{(k)'}, r^{(k)**} = r^{(k)*} + r^{(k)'} \end{aligned} \right) \quad (73)$$

$$\therefore u_p^{(k)**} = HP_p[u^{(k)**}] - r_p^{(k)**} \mathbf{D}_p^{(k)} \nabla_p P^* = HP_p[u^{(k)*} + u^{(k)'}] - (r_p^{(k)*} + r_p^{(k)'}) \mathbf{D}_p^{(k)} \nabla_p (P^\circ + P') \quad (74)$$

$$\therefore \begin{cases} u_p^{(k)'} = HP_p[u^{(k)'}] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' - r_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P^\circ - r_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P' \\ \rho^{(k)'} = C_p^{(k)} P' \\ r_p^{(k)'} = -R_p^{(k)} \left( \frac{r_p^{(k)*} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p [(r^{(k)*} \rho^{(k)\circ} u^{(k)'}) \cdot \mathbf{S} + r^{(k)*} U^{(k)*} \rho^{(k)'}] \right) \end{cases} \quad (75)$$

Condition:

$$\sum_k \{r_p^{(k)'}\} = \text{RESG}_p \quad (76)$$

$$\therefore \sum_k \left\{ -R_p^{(k)} \left[ \frac{r_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_p^{(k)} P'] + \Delta_p [r^{(k)*} \rho^{(k)\circ} (\mathbf{HP}[\mathbf{u}^{(k)'}] - r^{(k)*} \mathbf{D}^{(k)} \nabla P' - r^{(k)'} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] + \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] \right] \right\} = \text{RESG}_p \quad (77)$$

$$\sum_k \left\{ R_p^{(k)} \left[ \frac{r_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_p^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right] \right\} =$$

$$\therefore - \sum_k \left\{ R_p^{(k)} \left[ \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] + \Delta_p [r^{(k)*} \rho^{(k)\circ} (\mathbf{HP}[\mathbf{u}^{(k)'}] - r^{(k)'} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right] \right\} - \text{RESG}_p \quad (78)$$

Approximation:

Neglect:  $\mathbf{HP}[\mathbf{u}^{(k)'}], r^{(k)'} \mathbf{D}^{(k)} \nabla P'$

$$\Rightarrow \mathbf{u}_p^{(k)'} = -r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla P' \quad (79)$$

Approximate Equation:

$$\sum_k \left\{ R_p^{(k)} \left[ \frac{r_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P'_p + \Delta_p [r^{(k)*} U^{(k)*} C_p^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)} \nabla P') \cdot \mathbf{S}] \right] \right\} =$$

$$\Rightarrow - \sum_k \left\{ R_p^{(k)} \left[ \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] \right] \right\} - \text{RESG}_p \quad (80)$$

Second Predictor:

$$r_p^{(k)***} = H_p^{**} [r^{(k)**}] \quad (81)$$

$$\mathbf{u}_p^{(k)***} = \mathbf{HP}_p^{**} [\mathbf{u}^{(k)**}] - r_p^{(k)***} \mathbf{D}_p^{(k)**} \nabla P^* \quad (82)$$

Second Corrector:

$$\left( \mathbf{u}^{(k)'} , P'', \rho^{(k)'}, r^{(k)'} \right) \left( \begin{aligned} \mathbf{u}^{(k)****} &= \mathbf{u}^{(k)***} + \mathbf{u}^{(k)'}, P^* = P^* + P'', \\ \rho^{(k)****} &= \rho^{(k)**} + \rho^{(k)'}, r^{(k)****} = r^{(k)***} + r^{(k)'} \end{aligned} \right) \quad (83)$$

$$\therefore \mathbf{u}_p^{(k)****} = \mathbf{HP}_p^{**} [\mathbf{u}^{(k)****}] - r_p^{(k)****} \mathbf{D}_p^{(k)**} \nabla P^* \quad (84)$$

$$\mathbf{u}_p^{(k)***} = \mathbf{HP}_p^{**} [\mathbf{u}^{(k)**}] - r_p^{(k)***} \mathbf{D}_p^{(k)**} \nabla P^* \quad (85)$$

$$\left\{ \begin{aligned} \mathbf{u}_p^{(k)'} &= \mathbf{HP}_p^{**} [\mathbf{u}^{(k)****} - \mathbf{u}^{(k)**}] - (r_p^{(k)***} + r_p^{(k)'}) \mathbf{D}_p^{(k)**} \nabla P^* + r_p^{(k)**} \mathbf{D}_p^{(k)**} \nabla P^* \\ &= (\mathbf{HP}_p^{**} [\mathbf{u}^{(k)****} - \mathbf{u}^{(k)**} + \mathbf{u}^{(k)'}] - r_p^{(k)'} \mathbf{D}_p^{(k)**} \nabla P'' - r_p^{(k)***} \mathbf{D}_p^{(k)**} \nabla P'' - r_p^{(k)'} \mathbf{D}_p^{(k)**} \nabla P^*) \\ \therefore \rho^{(k)'} &= C_p P'' \\ r_p^{(k)'} &= -R_p^{(k)**} \left( \frac{r_p^{(k)***} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p [(r^{(k)***} \rho^{(k)*} \mathbf{u}^{(k)'}) \cdot \mathbf{S} + r^{(k)***} U^{(k)***} \rho^{(k)'}] \right) \end{aligned} \right. \quad (86)$$

Condition:

$$\sum_k \{r_p^{(k)*}\} = 1 - \sum_k r_p^{(k)***} = \text{RESG}_p \quad (87)$$

$$\sum_k \left\{ -R_p^{(k)**} \left[ \begin{aligned} & \frac{r_p^{(k)***} \Omega_p C_\rho^{(k)}}{\delta t} P_p'' + \Delta_p [r^{(k)***} U^{(k)***} C_\rho^{(k)} P_p''] + \\ & \Delta_p \left[ r^{(k)***} \rho^{(k)*} \left( \mathbf{H} P^{**} [\mathbf{u}^{(k)***} - \mathbf{u}^{(k)**} + \mathbf{u}^{(k)*}] \right. \right. \\ & \quad \left. \left. - r^{(k)*} \mathbf{D}^{(k)**} \nabla P'' - r^{(k)***} \mathbf{D}^{(k)**} \nabla P'' \right) \cdot \mathbf{S} \right] \\ & + \frac{(r_p^{(k)***} \rho_p^{(k)*}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [r^{(k)***} \rho^{(k)*} U^{(k)***}] \end{aligned} \right] \right\} = \text{RESG}_p \quad (88)$$

$$\begin{aligned} & \sum_k \left\{ R_p^{(k)**} \left[ \begin{aligned} & \frac{r_p^{(k)***} \Omega_p C_\rho^{(k)}}{\delta t} P_p'' + \Delta_p [r^{(k)***} U^{(k)***} C_\rho^{(k)} P_p''] - \\ & \Delta_p [r^{(k)***} \rho^{(k)*} (r^{(k)***} \mathbf{D}^{(k)**} \nabla P'') \cdot \mathbf{S}] \end{aligned} \right] \right\} = \\ \therefore & \left\{ -\sum_k R_p^{(k)**} \left[ \begin{aligned} & \frac{(r_p^{(k)***} \rho_p^{(k)*}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [r^{(k)***} \rho^{(k)*} U^{(k)***}] + \\ & \Delta_p \left[ r^{(k)***} \rho^{(k)*} \left( \mathbf{H} P^{**} [\mathbf{u}^{(k)***} - \mathbf{u}^{(k)**} + \mathbf{u}^{(k)*}] \right. \right. \\ & \quad \left. \left. - r^{(k)*} \mathbf{D}^{(k)**} \nabla P'' \right) \cdot \mathbf{S} \right] \end{aligned} \right] \right\} - \text{RESG}_p \quad (89) \end{aligned}$$

Approximation:

Neglect:  $\mathbf{H} P^{**} [\mathbf{u}^{(k)***} - \mathbf{u}^{(k)**} + \mathbf{u}^{(k)*}], r^{(k)*} \mathbf{D}^{(k)**} \nabla P''$

$$\Rightarrow \mathbf{u}^{(k)*} = -r_p^{(k)*} \mathbf{D}^{(k)**} \nabla P_p'' \quad (90)$$

Approximate Equation:

$$\begin{aligned} & \sum_k \left\{ R_p^{(k)**} \left[ \begin{aligned} & \frac{r_p^{(k)***} \Omega_p C_\rho^{(k)}}{\delta t} P_p'' + \Delta_p [r^{(k)***} U^{(k)***} C_\rho^{(k)} P_p''] - \\ & \Delta_p [r^{(k)***} \rho^{(k)*} (r^{(k)***} \mathbf{D}^{(k)**} \nabla P'') \cdot \mathbf{S}] \end{aligned} \right] \right\} = \\ & -\sum_k \left\{ R_p^{(k)**} \left[ \begin{aligned} & \frac{(r_p^{(k)***} \rho_p^{(k)*}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p [r^{(k)***} \rho^{(k)*} U^{(k)***}] \end{aligned} \right] \right\} - \text{RESG}_p \quad (91) \end{aligned}$$

**A Global GCBA-PISO Iteration**

- 
- Solve implicitly for  $r^{(k)}$ .
  - Solve implicitly for  $\mathbf{u}^{(k)}$  using the old pressure and density fields.
  - Calculate the  $\mathbf{D}^{(k)}$  fields.
  - Solve the pressure correction equation.
  - Correct  $\mathbf{u}^{(k)}$ ,  $P$ ,  $r^{(k)}$ , and  $\mathbf{r}^{(k)}$ .
  - Solve implicitly the energy equation and update the density fields.
  - Solve explicitly for  $r^{(k)}$ .
  - Solve the momentum equations explicitly and calculate the  $\mathbf{D}^{(k)}$  fields.
  - Solve the pressure correction equation.
  - Correct  $\mathbf{u}^{(k)}$ ,  $P$ ,  $r^{(k)}$ , and  $\mathbf{r}^{(k)}$ .
  - Return to step one and iterate until convergence
- 

### The GCBA following SIMPLEX (GCBA-SIMPLEX): Symbolic Form

Predictor:

$$\mathbf{r}_p^{(k)*} = \mathbf{H}_p[\mathbf{r}^{(k)*}] \quad (92)$$

$$\mathbf{u}_p^{(k)*} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)*}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^\circ \quad (93)$$

Corrector:

$$\left( \mathbf{u}^{(k)'} , P', \rho^{(k)'}, \mathbf{r}^{(k)'} \left\{ \begin{array}{l} \mathbf{u}^{(k)**} = \mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}, P^* = P^\circ + P', \\ \rho^{(k)*} = \rho^{(k)\circ} + \rho^{(k)'}, \mathbf{r}^{(k)**} = \mathbf{r}^{(k)*} + \mathbf{r}^{(k)'} \end{array} \right. \right) \quad (94)$$

$$\therefore \mathbf{u}_p^{(k)**} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)**}] - \mathbf{r}_p^{(k)**} \mathbf{D}_p^{(k)} \nabla_p P^* = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)*} + \mathbf{u}^{(k)'}] - (\mathbf{r}_p^{(k)*} + \mathbf{r}_p^{(k)'}) \mathbf{D}_p^{(k)} \nabla_p (P^\circ + P') \quad (95)$$

$$\therefore \left\{ \begin{array}{l} \mathbf{u}_p^{(k)'} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P^\circ - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P' \\ \rho^{(k)'} = C_p^{(k)} P' \\ \mathbf{r}_p^{(k)'} = -\mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{u}^{(k)'}), \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} \rho^{(k)'} \right] \right) \end{array} \right. \quad (96)$$

Condition:

$$\sum_k \{ \mathbf{r}_p^{(k)'} \} = \text{RESG}_p \quad (97)$$

$$\therefore \sum_k \left\{ -\mathbf{R}_p^{(k)} \left[ \frac{\mathbf{r}_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} P' + \Delta_p \left[ \mathbf{r}^{(k)*} \mathbf{U}^{(k)*} C_p^{(k)} P' \right] + \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} (\mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P') \mathbf{S} \right] + \frac{(\mathbf{r}_p^{(k)*} \rho_p^{(k)\circ}) - (\mathbf{r}_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ (\mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)*}) \right] \right] \right\} = \text{RESG}_p \quad (98)$$

Approximation:

Neglect:  $\mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p P'$ , and let

$$\mathbf{u}_p^{(k)'} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' = -\mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)SX} \nabla_p P' \quad (99)$$

$$\Rightarrow -r_p^{(k)*} \mathbf{D}_p^{(k)SX} \nabla_p P' = \mathbf{H}P_p [-r^{(k)*} \mathbf{D}^{(k)SX} \nabla P'] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P' \quad (100)$$

Assume that the pressure difference local to the velocity is representative of all pressure

differences i.e.  $\mathbf{H}P_p [-r^{(k)*} \mathbf{D}^{(k)SX} \nabla P'] = -(\nabla_p P') \mathbf{H}P_p [r^{(k)*} \mathbf{D}^{(k)SX}]$ , thus:

$$-r_p^{(k)*} \mathbf{D}_p^{(k)SX} (\nabla_p P') = -(\nabla_p P') \mathbf{H}P_p [r^{(k)*} \mathbf{D}^{(k)SX}] - r_p^{(k)*} \mathbf{D}_p^{(k)} (\nabla_p P') \quad (101)$$

$$\Rightarrow r_p^{(k)*} \mathbf{D}_p^{(k)SX} = \mathbf{H}P_p [r^{(k)*} \mathbf{D}^{(k)SX}] + r_p^{(k)*} \mathbf{D}_p^{(k)} \quad (102)$$

Approximate Equation:

$$\begin{aligned} \Rightarrow \sum_k \left\{ R_p^{(k)} \left( \frac{r_p^{(k)*} \Omega_p C_\rho^{(k)}}{\delta t} P' + \Delta_p [r^{(k)*} U^{(k)*} C_\rho^{(k)} P'] - \Delta_p [r^{(k)*} \rho^{(k)\circ} (r^{(k)*} \mathbf{D}^{(k)SX} \nabla P') \cdot \mathbf{S}] \right) \right\} = \\ - \sum_k \left\{ R_p^{(k)} \left( \frac{(r_p^{(k)*} \rho_p^{(k)\circ}) - (r_p^{(k)} \rho_p^{(k)})^{Old}}{\delta t} \Omega_p + \Delta_p [(r^{(k)*} \rho^{(k)\circ} U^{(k)*})] \right) \right\} - \text{RESG}_p \end{aligned} \quad (103)$$

### A Global GCBA-SIMPLEX Iteration

- 
- Solve implicitly for  $r^{(k)}$ .
  - Solve implicitly for  $\mathbf{u}^{(k)}$ , using the old pressure and density fields.
  - Calculate the  $\mathbf{D}^{(k)}$  fields.
  - Solve implicitly for the  $\mathbf{D}^{(k)SX}$  fields.
  - Solve the pressure correction equation using these  $\mathbf{D}^{(k)SX}$  fields.
  - Correct  $\mathbf{u}^{(k)}$ ,  $P$ ,  $r^{(k)}$ , and  $\rho^{(k)}$ .
  - Solve implicitly the energy equations and update the density fields.
  - Return to the first step and iterate until convergence.
- 

### The GCBA following SIMPLEX (GCBA-SIMPLEX): Symbolic Form

First Predictor:

$$r_p^{(k)*} = H_p [r^{(k)*}] \quad (104)$$

Calculate the coefficients of the momentum equations.

First Corrector:

$$\left( \mathbf{u}^{(k)*}, P', \rho^{(k)*}, r^{(k)*} \right) \left( \begin{aligned} \mathbf{u}^{(k)*} &= \mathbf{u}^{(k)\circ} + \mathbf{u}^{(k)*'}, P^* = P^\circ + P', \\ \rho^{(k)*} &= \rho^{(k)\circ} + \rho^{(k)*'}, r^{(k)*} = r^{(k)*} + r^{(k)*'} \end{aligned} \right) \quad (105)$$

$$\therefore \mathbf{u}_p^{(k)*} = \mathbf{H}P_p [\mathbf{u}^{(k)*}] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^* = \mathbf{H}P_p [\mathbf{u}^{(k)\circ} + \mathbf{u}^{(k)*'}] - (r_p^{(k)*} + r_p^{(k)*'}) \mathbf{D}_p^{(k)} \nabla_p (P^\circ + P') \quad (106)$$

$$\mathbf{u}_p^{(k)\circ} = \mathbf{H}P_p [\mathbf{u}^{(k)\circ}] - r_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p P^\circ \quad (107)$$

$$\therefore \begin{cases} \mathbf{u}_p^{(k)'} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)'}] - \mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p \mathbf{P}' - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p \mathbf{P}^\circ - \mathbf{r}_p^{(k)'} \mathbf{D}_p^{(k)} \nabla_p \mathbf{P}' \\ \rho^{(k)'} = C_p^{(k)} \mathbf{P}' \\ \mathbf{r}_p^{(k)'} = -\mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p}{\delta t} \rho_p^{(k)'} + \Delta_p \left[ \left( \mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{u}^{(k)'} \right) \cdot \mathbf{S} + \mathbf{r}^{(k)*} \mathbf{U}^{(k)\circ} \rho^{(k)'} \right] \right) \end{cases} \quad (108)$$

Condition:

$$\sum_k \{\mathbf{r}_p^{(k)'}\} = \text{RESG}_p \quad (109)$$

$$\therefore \sum_k \left\{ -\mathbf{R}_p^{(k)} \left[ \frac{\mathbf{r}_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} \mathbf{P}' + \Delta_p \left[ \mathbf{r}^{(k)*} \mathbf{U}^{(k)\circ} C_p^{(k)} \mathbf{P}' \right] + \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla \mathbf{P}' - \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla \mathbf{P}' \right) \cdot \mathbf{S} \right] + \frac{\left( \mathbf{r}_p^{(k)*} \rho_p^{(k)\circ} \right) - \left( \mathbf{r}_p^{(k)} \rho_p^{(k)} \right)^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ \left( \mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)\circ} \right) \right] \right] \right\} = \text{RESG}_p \quad (110)$$

$$\begin{aligned} & \sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} \mathbf{P}' + \Delta_p \left[ \mathbf{r}^{(k)*} \mathbf{U}^{(k)\circ} C_p^{(k)} \mathbf{P}' \right] - \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla \mathbf{P}' \right) \cdot \mathbf{S} \right] \right) \right\} = \\ & \therefore - \sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{\left( \mathbf{r}_p^{(k)*} \rho_p^{(k)\circ} \right) - \left( \mathbf{r}_p^{(k)} \rho_p^{(k)} \right)^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ \left( \mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)\circ} \right) \right] + \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}] - \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla \mathbf{P}' \right) \cdot \mathbf{S} \right] \right) \right\} - \text{RESG}_p \end{aligned} \quad (111)$$

Approximation:

Neglect:  $\mathbf{H}\mathbf{P}[\mathbf{u}^{(k)'}], \mathbf{r}^{(k)'} \mathbf{D}^{(k)} \nabla \mathbf{P}'$

$$\Rightarrow \mathbf{u}_p^{(k)'} = -\mathbf{r}_p^{(k)*} \mathbf{D}_p^{(k)} \nabla_p \mathbf{P}' \quad (112)$$

Approximate Equation:

$$\begin{aligned} & \sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{\mathbf{r}_p^{(k)*} \Omega_p C_p^{(k)}}{\delta t} \mathbf{P}' + \Delta_p \left[ \mathbf{r}^{(k)*} \mathbf{U}^{(k)\circ} C_p^{(k)} \mathbf{P}' \right] - \Delta_p \left[ \mathbf{r}^{(k)*} \rho^{(k)\circ} \left( \mathbf{r}^{(k)*} \mathbf{D}^{(k)} \nabla \mathbf{P}' \right) \cdot \mathbf{S} \right] \right) \right\} = \\ & \Rightarrow - \sum_k \left\{ \mathbf{R}_p^{(k)} \left( \frac{\left( \mathbf{r}_p^{(k)*} \rho_p^{(k)\circ} \right) - \left( \mathbf{r}_p^{(k)} \rho_p^{(k)} \right)^{\text{Old}}}{\delta t} \Omega_p + \Delta_p \left[ \left( \mathbf{r}^{(k)*} \rho^{(k)\circ} \mathbf{U}^{(k)\circ} \right) \right] \right) \right\} - \text{RESG}_p \end{aligned} \quad (113)$$

Second Predictor:

$$\mathbf{u}_p^{(k)**} = \mathbf{H}\mathbf{P}_p[\mathbf{u}^{(k)**}] - \mathbf{r}_p^{(k)**} \mathbf{D}_p^{(k)*} \nabla_p \mathbf{P}^* \quad (114)$$

Second Corrector:

No corrector stage.

**A Global MCBA-SIMPLEM Iteration**

- 
- Solve implicitly for  $r^{(k)}$ .
  - Calculate the  $\mathbf{D}^{(k)}$  fields based on values from the previous iteration.
  - Solve the pressure correction equation.
  - Correct  $\mathbf{u}^{(k)}$ ,  $P$ ,  $r^{(k)}$ , and  $\rho^{(k)}$ .
  - Calculate new  $\mathbf{HP}^{(k)}$  and  $\mathbf{D}^{(k)}$  fields.
  - Solve implicitly for  $\mathbf{u}^{(k)}$  using the new fields.
  - Solve implicitly the energy equations and update the density fields.
  - Return to the first step and iterate until convergence.
- 

### The Expanded Form of the Pressure-Correction Equation

If  $r^{(k)}$ ,  $\mathbf{U}^{(k)}$  and  $\rho^{(k)}$  denote values from the previous iteration or from a previous corrector step, the pressure correction equation, applicable to all algorithms, becomes

$$\sum_k \left\{ R_p^{(k)} \left( \frac{r_p^{(k)} C_p^{(k)} \Omega_p}{\delta t} P'_p + \Delta_p [r^{(k)} \mathbf{U}^{(k)} C_p^{(k)} P'] - \Delta_p [r^{(k)} \rho^{(k)} (r^{(k)} \mathbf{D}^{(k)} \nabla(P')) \cdot \mathbf{S}] \right) \right\} =$$

$$- \sum_{k=1}^{nphase} \left\{ R_p^{(k)} \left( \frac{(r_p^{(k)} \rho_p^{(k)}) - (r_p^{(k)} \rho_p^{(k)})^{Old}}{\delta t} \Omega_p + \Delta_p [(r^{(k)} \rho^{(k)} \mathbf{U}^{(k)})] \right) \right\} - RESG_p \quad (115)$$

The discretization of the above equation yields

$$A_P^{P'} P'_p = A_E^{P'} P'_e + A_W^{P'} P'_w + A_N^{P'} P'_n + A_S^{P'} P'_s + B_P^{P'} \quad (116)$$

where

$$A_E^{P'} = \sum_k R_p^{(k)} [\Gamma_e^{(k)} + (r^{(k)} C_p^{(k)})_e] \| -U_e^{(k)}, 0 \| \quad A_W^{P'} = \sum_k R_p^{(k)} [\Gamma_w^{(k)} + (r^{(k)} C_p^{(k)})_w] \| -U_w^{(k)}, 0 \|$$

$$A_N^{P'} = \sum_k R_p^{(k)} [\Gamma_n^{(k)} + (r^{(k)} C_p^{(k)})_n] \| -U_n^{(k)}, 0 \| \quad A_S^{P'} = \sum_k R_p^{(k)} [\Gamma_s^{(k)} + (r^{(k)} C_p^{(k)})_s] \| -U_s^{(k)}, 0 \|$$

$$A_P^{P'} = A_E^{P'} + A_W^{P'} + A_N^{P'} + A_S^{P'} +$$

$$\sum_k R_p^{(k)} \left( \frac{r_p^{(k)} K_p^{(k)} \Omega_p}{\delta t} + (r^{(k)} C_p^{(k)})_e U_s^{(k)} + (r^{(k)} C_p^{(k)})_w U_w^{(k)} + (r^{(k)} C_p^{(k)})_n U_n^{(k)} + (r^{(k)} C_p^{(k)})_s U_s^{(k)} \right)$$

$$B_P^{P'} = - \sum_k R_p^{(k)} \left\{ \frac{(r_p^{(k)} \rho_p^{(k)} - r_p^{(k)old} \rho_p^{(k)old})}{\delta t} \Omega_p \right. \\ \left. + (r_e^k \rho_e^{(k)} U_s^{(k)} + r_w^k \rho_w^{(k)} U_w^{(k)} + r_n^k \rho_n^{(k)} U_n^{(k)} + r_s^k \rho_s^{(k)} U_s^{(k)}) \right\} - RESG_p \quad (117)$$

Following the calculation of the pressure correction field,  $\mathbf{u}_p^{(k)'}$ ,  $\rho_p^{(k)'}$ , and  $r_p^{(k)'}$  are obtained

using the following equations

$$\begin{aligned}
\mathbf{u}_P^{(k)'} &= -r^{(k)} \mathbf{D}_P^{(k)} \nabla_P (P') \\
\rho^{(k)'} &= C_\rho^{(k)} P' \\
r_P^{(k)'} &= -R_P^{(k)} \left( \frac{r_P^{(k)} \Omega_P}{\delta t} \rho_P^{(k)'} + \Delta_P \left[ \left( r^{(k)} \rho^{(k)} \mathbf{u}^{(k)'} \right) \cdot \mathbf{S} + r^{(k)} U^{(k)} \rho^{(k)'} \right] \right)
\end{aligned} \tag{118}$$

## The Multi-Grid Strategy

Similar to other iterative methods, the rate of convergence of the solution method described above does not scale linearly with the grid size, rather the convergence rate decreases more drastically as the number of grid points increases. This behavior is attributed to the speed at which the iterative solver transports the boundary information across the domain (e.g. with SOR one grid point per iteration). Since information has to travel back and forth several times to achieve convergence, acceleration of the outer iterations through the use of multi-grid methods is essential when solving over large grids. The idea underlying the multi-grid strategy is to use progressively coarser grids to accelerate the convergence rate. In mathematical terms the low-frequency error components in the finest grid appear on coarser grids as high-frequency Fourier mode that can be resolved efficiently by iterative relaxation solvers. In the present work, this strategy is adopted to accelerate convergence and thereby reduce the overall computational cost. The method used is the FMG-FAS method [56]. For a review of Multi-grid methods the reader is referred, among others, to [56,57], therefore it is sufficient here to give a general description of the method used.

The multi-grid algorithm adopted in this work can be summarized as follows. Starting with the fine mesh, the coarser grid cells are generated through agglomeration of four finer grid cells, two in each direction. On the other hand, if a finer grid is required, subdividing the coarser grid control volume into four control volumes, again two in each direction, generates its control volumes. With the FMG cycle, the algorithm starts at the coarsest level, where the solution is first computed; this solution is interpolated onto the next finer mesh, where it is

used as initial guess. This stage is called the prolongation stage (see Fig. 1(b)). Then iterations are performed on the fine mesh and the solution is transferred back to the coarser mesh by applying a restriction operator. In order to obtain the same approximation on each level, a forcing term is added to the discrete conservation equations on the coarser grid. This term represents the truncation error on the coarse grid with respect to the fine grid. After performing a number of iterations on the coarse mesh, the solution is transferred back to the finer mesh in the form of a correction and a number of iterations are performed on the finer grid to smooth the fields. This process is continued until a converged solution on the fine mesh is obtained (see Fig. 1(c)). Then the solution is extrapolated to correct the finer mesh fields, followed by a number of smoother iterations on the finer mesh and the process repeated until convergence is reached on the desired finest mesh. This strategy has been applied to both incompressible and compressible supersonic multi-fluid flows and good savings have been realized as will be shown in the results section.

In the restriction step the coarse grid variables are computed from the fine grid values as:

$$\tilde{\phi}_C = \frac{1}{4} \sum_{i=1..4} (\phi_{F_i} + \nabla \phi_{F_i} \cdot \mathbf{d}_{F_i C}) \quad (119)$$

while in the prolongation step the fine grid corrections are computed from the coarse grid values as

$$\phi'_{F_i} = \phi'_C + \nabla \phi'_C \cdot \mathbf{d}_{CF_i} \quad (120)$$

where  $\mathbf{d}$  is the position vector connecting points C and  $F_i$  and  $\phi'_C$  given by

$$\phi'_C = \phi_C - \tilde{\phi}_C \quad (121)$$

The special character of the volume fraction and k- $\epsilon$  equations necessitates modification to the prolongation procedure as described next.

While extrapolating the volume fraction field from the coarse to the fine grid, the prolongation operator may yield negative volume fraction values or values that are greater than one. Such unphysical values are detrimental to the overall convergence rate and may

cause divergence. To circumvent this problem, a simple yet very effective treatment is adopted: Once the  $r$ -values are extrapolated, a check is performed to make sure they are within bounds. If any of the  $r$ -values is found to be unbounded, the  $r$ -phasic volume fraction equation is solved starting with the interpolated values until all of the  $r$ -values are within the set bounds. Typically less than 10 iterations are needed. This treatment has been found to be very effective and to preserve the convergence acceleration rate. The practices of solving the volume fraction equations only on the fine grid or forcing the extrapolated unbounded values to be within the set bounds or discarding corrections that result in unbounded values [57] proved to be ineffective and slowed the convergence rate considerably.

For the  $k$ - $\epsilon$  turbulence model, the treatment suggested by Cornelius et al. [58] is adopted. This approach is based on the observation that the application of wall functions to the coarse grids would lead to unphysical values because of the relatively large distance between the wall and the boundary cell center. Thus, at wall boundaries the restricted fine grid values of  $k$  and  $\epsilon$  are held constant, and hence no corrections are calculated. In order to satisfy the realizability constraint, the restricted turbulence properties and prolonged correction values are modified accordingly.

In addition to the FMG strategy, the PG approach is also tested. This approach differs from the FMG method in that the solution moves in one direction from the coarse to the fine grids with the initial guess on level  $n+1$  obtained by interpolation from the converged solution on level  $n$  (Fig. 1(b)). As such, the acceleration over the SG method obtained with this approach is an indication of the effect of initial guess on convergence.

## Results and Discussion

The performance of the various multi-fluid Geometric Conservation Based Algorithms is assessed in this section by presenting solutions to several one and two-dimensional two-phase

flow problems. Results are presented in terms of the CPU-time needed to converge the solution to a set level and of the convergence history. Moreover, solutions are obtained for a number of grids in order to assess the performance of the various algorithms with increasing grid density. For the two-dimensional problems, in addition to the CPU-time needed to solve a problem using the single grid method (SG), the CPU-time needed using different solution strategies is also displayed. These include the prolongation scheme (PG) in which the solution from the next coarser grid is used to provide the initial field, and the full multi-grid method (FMG). Results are compared against available experimental data and/or numerical/theoretical values. The residual of a variable  $\phi$  at the end of an outer iteration is defined as:

$$\text{RES}_{\phi}^{(k)} = \sum_{c.v} \left| A_p \phi_p^{(k)} - \sum_{\text{all } p \text{ neighbours}} A_{nb} \phi_{nb}^{(k)} - B_p^{(k)} \right| \quad (122)$$

For global mass conservation, the imbalance in mass is defined as:

$$\text{RES}_c = \sum_k \sum_{c.v.} \left| \frac{(r_p^{(k)} \rho_p^{(k)}) - (r_p^{(k)} \rho_p^{(k)})^{\text{Old}}}{\delta t} \Omega - \Delta_p [r^{(k)} \rho^{(k)} \mathbf{u}^{(k)} \cdot \mathbf{S}] - r^{(k)} \dot{M}^{(k)} \right| \quad (123)$$

All residuals are normalized by their respective inlet fluxes. Computations are terminated when the maximum normalized residual of all variables, drops below a very small number  $\epsilon_s$ . For a given problem, the same value of  $\epsilon_s$  is used with all algorithms. In general, it is found that requiring the overall mass residuals to be satisfied to within  $\epsilon_s$  is a very stringent requirement and the last to be fulfilled. This is why these residuals are the ones presented here and used to compare the performance of the various algorithms. In all problems, the first phase represents the continuous phase (denoted by a superscript (c)), which must be fluid, and the second phase is the disperse phase (denoted by a superscript (d)), which may be solid or fluid. Unless otherwise specified the HR SMART scheme is used in all computations reported in this study. For a given problem, all results are generated starting from the same

initial guess. Moreover, it should be stated that in iterative techniques, different initial guesses might require different computational efforts.

## **One-dimensional two-phase validation problems**

Due to the large number of parameters affecting the performance of the various multi-phase Geometric Conservation Based Algorithms and to allow a thorough testing of these algorithms, eight one-dimensional two-phase flow problems are considered. These problems can be broadly classified as: (i) horizontal particle transport, and (ii) vertical particle transport. Results are presented in terms of the convergence history and the CPU-time needed to converge the solution to a set level. Predictions are compared against available numerical/theoretical values.

Despite its geometric simplicity, the one-dimensional particle transport problem can represent a wide range of physical conditions. The effects of grid refinement on accuracy and convergence are studied by solving the problems on four grid systems of sizes 20, 40, 80, and 160 control volumes with  $\epsilon_s$  assigned the value of  $10^{-8}$ .

Many runs were performed so as to set the control parameters of each algorithm near optimum values. To allow a comparative assessment of performance, the CPU times are reported in the form of charts. Moreover, all CPU times are normalized by the time needed by GCBA-SIMPLE to reach the set residuals on the coarsest grid.

### ***Horizontal particle transport***

The physical situation is depicted in Fig. 1(d). Depending on the set densities, it represents either the steady flow of solid particles suspended in a free stream of air or the steady flow of air bubbles in a stream of water. The slip between the phases determines the drag, which is the sole driving force for the particle-bubble/air-water motion ( $g=0$ ). In the suspension, the inter-particle/bubble forces are neglected. Diffusion within both phases is set to zero while the inter-phase drag force is calculated as:

$$I_M^{(c)} = \frac{3}{8} \frac{C_D}{r_p} r^{(d)} \rho^{(c)} V_{\text{slip}} (\mathbf{u}^{(d)} - \mathbf{u}^{(c)}) \quad (124)$$

$$I_M^{(d)} = -\frac{3}{8} \frac{C_D}{r_p} r^{(d)} \rho^{(c)} V_{\text{slip}} (\mathbf{u}^{(d)} - \mathbf{u}^{(c)}) \quad (125)$$

$$V_{\text{slip}} = \|\mathbf{u}^{(d)} - \mathbf{u}^{(c)}\| \quad (126)$$

The drag coefficient,  $C_D$ , is set to 0.44. Since phasic diffusion is neglected, the GCBA-SIMPLEST and GCBA-PRIME becomes identical and reference will be made to GCBA-SIMPLEST only. The task is to calculate the particle/bubble-velocity distribution as a function of position. If the flow field is extended far enough (here computations are performed over a length of  $L=2\text{m}$ ), the particle/bubble and fluid phases are expected to approach an equilibrium velocity given by:

$$\mathbf{U}_{\text{equilibrium}} = r_{\text{inlet}}^{(c)} \mathbf{V}_{\text{inlet}}^{(c)} + r_{\text{inlet}}^{(d)} \mathbf{V}_{\text{inlet}}^{(d)} \quad (127)$$

### Problem 1: Dilute gas-solid flow

The steady flow of dilute particles suspended in a free stream of air is studied first. At inlet, the air and particle velocities are 5 m/s and 1 m/s, respectively. The physical properties of the two phases are:  $\rho^{(d)}/\rho^{(c)} = 2000$ ,  $r_p = 1 \text{ mm}$ ,  $r_{\text{inlet}}^{(d)} = 10^{-5}$ . Due to the dilute concentration of the particles, the free stream velocity is more or less unaffected by their presence and the equilibrium velocity is nearly equal to the inlet free stream velocity. Based on this observation, Morsi and Alexander [59] obtained the following analytical solution for the particle velocity  $\mathbf{u}^{(d)}$  as a function of the position  $x$  and the properties of the two phases:

$$\text{Ln} \left[ \mathbf{V}_{\text{inlet}}^{(c)} - \mathbf{u}^{(d)} \right] + \frac{\mathbf{V}_{\text{inlet}}^{(c)}}{\mathbf{V}_{\text{inlet}}^{(c)} - \mathbf{u}^{(d)}} = \frac{3}{8} \frac{\rho^{(c)}}{\rho^{(d)}} \frac{C_D}{r_p} x + \text{Ln} \left[ \mathbf{V}_{\text{inlet}}^{(c)} - \mathbf{V}_{\text{inlet}}^{(d)} \right] + \frac{\mathbf{V}_{\text{inlet}}^{(c)}}{\mathbf{V}_{\text{inlet}}^{(c)} - \mathbf{V}_{\text{inlet}}^{(d)}} \quad (128)$$

This case is of particular importance since the flow situation has an exact solution. As shown in Fig. 2(a) the predicted particle velocity distribution falls on top of the analytical solution given by Eq. (128), which is an indication of the accuracy of the numerical procedure. The convergence histories of the various GCBA over the four grid networks used are displayed in Figs. 2(b)-2(h). For all algorithms, the required number of iterations increases as the grid size

increases, with PISO (Fig. 2(b)) requiring the minimum and SIMPLEST/PRIME (Fig. 2(f)) the maximum number of iterations on all grids. The convergence histories of SIMPLE, SIMPLEC, SIMPLEMENT, and SIMPLEX (Figs. 2(c), 2(d), 2(e), and 2(g), respectively) are very similar with SIMPLEMENT (Fig. 2(e)) requiring the lowest number of iterations. The convergence paths of the various algorithms over a grid of size 80 C.V. are compared in Fig. 2(h) and the above observations are easily inferred from the figure.

### Problem 2: Dense gas-solid flow

The only difference between this case and the previous one is in the concentration of particles, which is set to  $r_{\text{inlet}}^{(d)} = 10^{-2}$ . Despite the low value of the inlet disperse phase volume fraction, the ratio of disperse phase and continuous phase mass loadings is large  $r^{(d)}\rho^{(d)} / r^{(c)}\rho^{(c)} = 20$ . Thus the disperse phase carries most of the inertia of the mixture. The equilibrium velocity in this case, as obtained from Eq. (127) is 4.96 m/s as compared to 4.99996 m/s in the previous case. Due to this slight difference between the inlet air velocity and the final equilibrium velocity, the free stream velocity may be assumed to be nearly constant and the variation in particle velocity can be obtained again from Eq. (128). The predicted air and particle velocity distributions are displayed in Fig. 3(a). The numerical and analytical particle velocity profiles are indistinguishable and fall on top of each other. Moreover, the slight decrease in the air velocity can be easily depicted. The convergence paths for all algorithms and over all grid systems used are displayed in Figs. 3(b)-3(h). In general, higher number of iterations is required to reach the desired level of convergence on a given grid as compared to the dilute case due to the increased importance of the inter-phase term. The general convergence trend is similar to that of the dilute problem with PISO requiring the minimum and SIMPLEST the maximum number of iterations. The SIMPLEMENT algorithm (Fig. 3(e)) is seen to require a slightly lower number of iterations on the finest grid as compared to SIMPLE (Fig. 3(c)), SIMPLEC (Fig. 3(d)), and SIMPLEX (Fig. 3(c)). As

depicted in Figs. 3(f) and 3(h), the performance of SIMPLEST/PRIME is poor as compared to other algorithms for the same reasons stated above.

### Problem 3: Dilute bubbly flow

For the same configuration displayed in Fig. 1(d), the continuous phase is considered to be water and the disperse phase to be air. The resulting flow is denoted in the literature by bubbly flow. With the exception of  $\rho^{(d)} / \rho^{(c)} = 10^{-3}$  and at inlet  $r_{\text{inlet}}^{(d)} = 0.1$ , other physical properties and inlet conditions are the same as those considered earlier. This is a strongly coupled problem and represents a good test for the numerical procedure and performance of the algorithms. The correct physical solution is that the bubble and continuous phase velocities both reach the equilibrium velocity of 4.6 m/s (Eq. (127)) in a distance too small to be correctly resolved by any of the grid networks used. Results for this case are presented in Fig. 4. Axial velocity distribution for both water and air are displayed in Fig. 4(a). As expected, both phases reach the equilibrium velocity of 4.6 m/s over a very short distance from the inlet section and remain constant afterward. The relative convergence characteristics of the various algorithms remain the same. However, all algorithms require larger number of iterations as compared to the dilute gas solid flow case due to the stronger coupling between the phases. Consistently, the PISO (Fig. 4(b)) and SIMPLEST/PRIME (Fig. 4(f)) algorithms need the lowest and highest number of iterations, respectively. As in the previous two cases, the convergence attributes of SIMPLE (Fig. 4(c)), SIMPLEC (Fig. 4(d)), SIMPLEM, and SIMPLEX (Fig. 4(g)) are very similar with SIMPLEM consistently requiring a lower number of iterations. The large difference in performance between SIMPLEST/PRIME and the remaining algorithms is clearly demonstrated in Fig. 4(h).

### Problem 4: Dense bubbly flow

The only difference between this case and the previous one is in the concentration of bubbles, which is set to  $r_{\text{inlet}}^{(d)} = 0.5$ . With such high value of void fraction, bubble coalescence may

occur. However, this is not accounted for here. The analytical solution is the same as in the previous case with the equilibrium velocity, as computed from Eq. (127), being 3 m/s. As depicted in Fig. 5(a), the equilibrium velocity obtained numerically is exact. With the exception of requiring higher number of iterations to reach the desired level of convergence, the performance of the various algorithms (Figs. 5(b)-5(h)) vary relatively in a manner similar to what was previously discussed and deemed redundant to be repeated.

### **CPU time: Horizontal particle transport**

The normalized CPU efforts required by the various algorithms over all grids are depicted in Fig. 6. The charts clearly show that the CPU time increases with increasing grid density. For the dilute gas-solid problem (Fig. 6(a)), it is hard to see any noticeable difference in the CPU times for SIMPLE, SIMPLEM, and SIMPLEX. The SIMPLEC and PISO algorithms require slightly lower and higher computational efforts, respectively, as compared to SIMPLE. The worst performance is for SIMPLEST which degenerates to PRIME in the absence of diffusion and results in a fully explicit solution scheme. For the dense gas-solid flow (Fig. 6(b)), the computational times needed by SIMPLE, SIMPLEC, SIMPLEM, and SIMPLEX are nearly identical. PISO, however, requires higher computational effort (50% more than SIMPLE on the finest meshes (80 and 160 C.V.)). The computational effort needed by SIMPLEST/PRIME is however the most extensive and is nearly 500% the one needed by SIMPLE on the finest mesh.

The normalized CPU time of SIMPLEST/PRIME for the bubbly flow problems (Figs. 6(c) and 6(d)) is lower than in the previous two problems due to a higher rate of increase in the time needed by other algorithms (the computational time of all algorithms has increased). The relative performance of the various algorithms is nearly as described earlier with the time required by of PISO, SIMPLE, SIMPLEC, and SIMPLEX being on average the same. The SIMPLEST/PRIME algorithm however, requires nearly 50% more time than SIMPLE, which

represents a noticeable improvement. The best performance for the dense bubbly flow problem is for SIMPLEM, which requires about 50% less effort on the finest mesh than SIMPLE.

### ***Vertical particle transport***

Here, the flow is in the vertical direction (Fig. 1(d)), the gravitational acceleration is assigned the constant value of  $g=10 \text{ m/s}^2$ , and the flow field is extended over a length of  $L=20\text{m}$ . For this situation, the velocities of the two phases do not reach an equilibrium value. Rather, the disperse phase equilibrates to a finite settling velocity relative to the continuous phase, at which the gravitational force balances the drag force [59]. As for the horizontal transport problems, the inter-particle/bubble forces are neglected. However, unlike the previous situation, diffusion in the continuous phase is retained. Moreover, the inter-phase drag force is calculated using Eqs. (124)-(126) and the drag coefficient,  $C_D$ , is considered to be particle Reynolds number dependent and calculated as:

$$C_D = \frac{24}{\text{Re}_p} + 0.44, \quad \text{Re}_p = \frac{2r_p V_{\text{slip}}}{\vartheta^{(c)}} \quad (129)$$

Since diffusion in the continuous phase is not neglected, GCBA-SIMPLEST and GCBA-PRIME are expected to behave differently.

### **Problem 5: Dilute gas-solid flow**

The material properties and boundary conditions considered for this case are given by:

$$\rho^{(d)} / \rho^{(c)} = 1000, \quad \vartheta^{(c)} = 10^{-5}, \quad r_p = 1 \text{ mm} \quad (130)$$

$$V_{\text{inlet}}^{(c)} = 100 \text{ m/s}, \quad V_{\text{inlet}}^{(d)} = 10 \text{ m/s}, \quad r_{\text{inlet}}^{(d)} = 10^{-5} \quad (131)$$

The large velocity boundary condition is used to ensure that the solid phase does not exit the inlet. The predicted air and particle velocity distributions depicted in Fig. 7(a) are in excellent agreement with similar predictions reported in [60]. As shown in Figs. 7(b)-7(h), the mass residuals tend to slightly increase at the beginning of the iterative process, stagnate over a number of iterations (this number increases with increasing grid size), and then decrease

rapidly to the desired level of convergence. This behavior is also true for the horizontal case discussed earlier and is attributed to the approximations introduced to the pressure correction equation especially with regard to neglecting second order correction terms, which may be important at the beginning of the iterative process. Once these neglected terms become unimportant, the rate of convergence increases drastically. Retaining these terms could improve the convergence rate but this has not been considered in this work.

As depicted in Fig. (7), the number of iterations needed to converge the solution to the desired level is very close to that needed in the similar horizontal transport case. This is equally true with regard to the relative performance of the various algorithms. The performance of SIMPLEST (Fig. 7(f)) and PRIME (Fig. 7(g)) is very close due to the fact that the implicitness introduced by the diffusion of the continuous phase does not seem to be that important. However, both require on the finest mesh almost 430% the number of iterations needed by SIMPLE. The number of outer iterations needed by SIMPLEX and SIMPLEC is very close to that of SIMPLE, while SIMPLEM entails lower number of iterations. Again, PISO requires the lowest number of iterations.

### **Problem 6: Dense gas-solid flows**

The material properties and boundary conditions are similar to the previous case with the exception of the particles' volume fraction, which is set to  $r_{\text{inlet}}^{(d)} = 10^{-2}$ . Predicted air and particle velocity profiles are displayed in Fig. 8(a) while mass residuals are presented in Figs. 8(b)-8(h). Higher number of iterations is needed in comparison with the dilute case due to the higher mass-loading ratio. Besides that, the convergence behavior is similar to the previous cases with SIMPLEST (Fig. 8(f)) and PRIME (Fig. 8(g)) requiring the highest number of iterations and PISO (Fig. 8(b)) the lowest number of iterations. The number of iterations needed by SIMPLE, SIMPLEC, and SIMPLEX (compare Figs. 8(c), 8(d), and 8(h)) is very

close. The SIMPLEM algorithm (Fig. 8(e)) needs about 25% less iterations than SIMPLE (Fig. 8(c)) on the finest mesh.

### Problem 7: Dilute bubbly flows

In this problem, the continuous phase is water and the disperse phase is air. With the exception of  $\rho^{(d)}/\rho^{(c)}$  set to  $10^{-3}$ ,  $U^{(d)} = U^{(c)} = 1$ , and  $r_{\text{inlet}}^{(d)}$  to 0.1 at inlet, other physical properties and inlet conditions are the same as those considered earlier. This is a very difficult problem to get convergence to unless the proper under-relaxation is used. By starting with relatively high under-relaxation factors, the number of iterations needed with all algorithms was found to be very high. In order to get feasible solutions, the under-relaxation factors during the first 20 iterations were set to 0.05 and then increased to the desired values. This was found to greatly improve the convergence rate and to generate solutions with nearly the same computational effort needed in the previous cases. In addition, this treatment has improved the performance of SIMPLEST and PRIME dramatically and has decreased their required number of iterations to values similar to those needed by other algorithms (Figs. 9(b)-9(h)). In fact, SIMPLEST and PRIME are performing slightly better than SIMPLE for this particular problem. Overall, none of the algorithms shows an outstanding superiority in performance over others.

### Problem 8: Dense bubbly flows

With the exception of setting  $r_{\text{inlet}}^{(d)}$  to 0.5, the physical situation, material properties, and boundary conditions are the same as in the previous problem. Results for the problem are presented in Fig. 10. It was possible to get feasible solutions (i.e. with reasonable computational time) only when under-relaxing by inertia (i.e. through the use of false time steps). For the results presented in Fig. 10, a time step ( $\Delta t$ ) of value  $10^{-4}$  s is used for the velocity field of the dispersed gas phase,  $\Delta t = 1$  s for the volume fractions, and  $\Delta t = 0.01$  s for

the velocity field of the liquid phase for the grids of sizes 20 and 40 C.V. and  $\Delta t = 0.05$  for all variables and for both phases with the dense grids (i.e. 80 and 160 C.V.). The predicted liquid and gas velocity distributions, which are in excellent accord with published data, are depicted in Fig. 10(a). The trend of convergence (Figs. 10(b)-10(h)) is different than what has been presented so far with all algorithms, except PISO, requiring nearly the same number of iterations and behaving in almost the same manner. It is also noticed that the number of iterations needed on the finest mesh is smaller than the numbers needed on the grids of sizes 40 and 80 C.V. Nevertheless, it was possible to obtain solutions with all GCBA algorithms.

### **CPU time: Vertical particle transport**

The normalized CPU times for the vertical particle transport problems are displayed in Fig. 11. As in the horizontal case, the CPU time increases with increasing grid density. For the gas-solid flow problems (Figs. 11(a) and 11(b)), the relative performance of the various algorithms is similar for both dilute and dense concentration of particles. For the dilute case (Fig. 11(a)), the efficiency of PRIME is slightly better than SIMPLEST (due to the use of an explicit algebraic-equation solver), both however are about four times more expensive than all other algorithms whose performance is very comparable (i.e. of the same order of magnitude) with SIMPLEM being the least expensive (7% less than SIMPLE on the finest mesh) and PISO the most expensive (9.5% more than SIMPLE on the finest mesh). The same is true for the dense gas-solid case (Fig. 11(b)) with the performance of SIMPLE, SIMPLEC, SIMPLEM, SIMPLEX, and PISO being closer.

For the vertical bubbly flows, a noticeable change in the normalized time chart (Figs. 11(c) and 11(d)) is depicted, with the performance of SIMPLEST and PRIME showing good improvements while the performance of the remaining algorithms deterioration. As depicted, the CPU times needed by the various algorithms are of the same order of magnitude with SIMPLEM being slightly more expensive.

By comparing the behavior of the various algorithms in all problems, it is clear that the performance of SIMPLE, SIMPLEC, SIMPLEMENT, SIMPLEX, and PISO is consistent and require, on average, the least computational effort. The performance of the SIMPLEST and PRIME algorithms was comparable to SIMPLE for upward bubbly flows only and they were, in general, the most expensive to use on all grids and for all physical situations presented here. Most importantly however, is the fact that all these algorithms can be used to predict multi-phase (in this case two-phase) flows.

## Two-dimensional two-phase validation problems

In this section, four two-dimensional two-phase flow problems are solved. The first two problems deal with incompressible turbulent flows while the last two problems are concerned with compressible flows.

### *Problem 1: Turbulent upward bubbly flow in a pipe*

Many experimental and numerical studies involving the prediction of radial phase distribution in turbulent upward air-water flow in a pipe have appeared in the literature [61-68]. These studies indicated that the lateral forces that most strongly affect the void distribution are the lateral lift force and the turbulent stresses. As such, in addition to the usual drag force, the lift force is considered as part of the interfacial force terms in the momentum equations. In the present work, the interfacial drag forces per unit volume are given by:

$$\left(\mathbf{I}_M^x\right)_D^{(c)} = -\left(\mathbf{I}_M^x\right)_D^{(d)} = 0.375 \frac{C_D}{r_p} \rho^{(c)} \mathbf{r}^{(d)} \cdot \mathbf{r}^{(c)} \mathbf{V}_{\text{slip}} \left(\mathbf{u}^{(d)} - \mathbf{u}^{(c)}\right) \quad (132)$$

$$\left(\mathbf{I}_M^y\right)_D^{(c)} = -\left(\mathbf{I}_M^y\right)_D^{(d)} = 0.375 \frac{C_D}{r_p} \rho^{(c)} \mathbf{r}^{(d)} \cdot \mathbf{r}^{(c)} \mathbf{V}_{\text{slip}} \left(\mathbf{v}^{(d)} - \mathbf{v}^{(c)}\right) \quad (133)$$

where  $r_p$  is the bubble radius. The drag coefficient  $C_D$  varies as a function of the bubble Reynolds and Weber numbers defined as:

$$\begin{cases} \text{Re}_p = 2 \frac{r_p}{v_l^{(c)}} V_{\text{slip}} \\ \text{We} = 4\rho^{(c)} \frac{r_p^2}{\sigma} V_{\text{slip}} \end{cases} \quad (134)$$

where  $\sigma$ , the surface tension, is assigned a value of 0.072 N/m for air-water systems. The drag coefficient is computed using the following correlations [69,70]:

$$\begin{cases} C_D = \frac{16}{\text{Re}_p} & \text{for } \text{Re}_p < 0.49 \\ C_D = \frac{20}{\text{Re}_p^{0.643}} & \text{for } 0.49 < \text{Re}_p < 100 \\ C_D = \frac{6.3}{\text{Re}_p^{0.385}} & \text{for } \text{Re}_p \gg 100 \\ C_D = \frac{8}{3} & \text{for } \text{Re}_p \gg 100 \text{ and } \text{We} > 8 \\ C_D = \frac{\text{We}}{3} & \text{for } \text{Re}_p \gg 100 \text{ and } \text{Re}_p > 2065.1 / \text{We}^{2.6} \end{cases} \quad (135)$$

Many investigators have considered the modeling of lift forces [69-71]. Based on their work, the following expressions are employed for the calculation of the interfacial lift forces per unit volume:

$$(\mathbf{I}_M)_L^{(c)} = -(\mathbf{I}_M)_L^{(d)} = C_1 \rho^{(c)} r^{(d)} (\mathbf{u}^{(d)} - \mathbf{u}^{(c)}) \cdot \nabla \mathbf{x} \mathbf{u}^{(c)} \quad (136)$$

where  $C_1$  is the interfacial lift coefficient calculated from:

$$C_1 = C_{1a} (1 - 2.78 \langle 0.2, r^{(d)} \rangle) \quad (137)$$

where  $\langle a, b \rangle$  denotes the minimum of  $a$  and  $b$  and  $C_{1a}$  is an empirical constant.

The effect of bubbles on the turbulent field is very important. In this work, turbulence is assumed to be a property of the continuous liquid phase (c) and is computed by solving Eqs.

(4) and (5) with  $\mathbf{I}_k^{(k)}$  and  $\mathbf{I}_\varepsilon^{(k)}$  given by [69]:

$$\mathbf{I}_k^{(c)} = \nabla \cdot \left[ \rho^{(c)} \left( \frac{\mathbf{v}_t^{(c)}}{\sigma_r} \right) \mathbf{k}^{(c)} \nabla \mathbf{r}^{(c)} \right] + \mathbf{r}^{(c)} P_b \quad (138)$$

$$\mathbf{I}_\varepsilon^{(c)} = \nabla \cdot \left[ \rho^{(c)} \left( \frac{\mathbf{v}_t^{(c)}}{\sigma_r} \right) \boldsymbol{\varepsilon}^{(c)} \nabla \mathbf{r}^{(c)} \right] + \mathbf{r}^{(c)} c_{1\varepsilon} P_b \frac{\boldsymbol{\varepsilon}^{(c)}}{\mathbf{k}^{(c)}} \quad (139)$$

where  $\sigma_r$  is the turbulent Schmidt number for volume fractions, and  $P_b$  is the production rate of  $k^{(c)}$  by drag due to the motion of the bubbles through the liquid and is given by:

$$P_b = \frac{0.375 C_b C_D \rho^{(c)} r^{(d)} r^{(c)} V_{slip}^2}{r_p} \quad (140)$$

In Eq. (140)  $C_b$  is an empirical constant representing the fraction of turbulence induced by bubbles that goes into large-scale turbulence of the liquid phase. Moreover, as suggested in [69], the flux representing the interaction between the fluctuating velocity and volume fraction is modeled via a gradient diffusion approximation and added as a source term in the continuity  $(\nabla \cdot (\rho^{(k)} D^{(k)} \nabla r^{(k)}))$  and momentum  $(\nabla \cdot (\rho^{(k)} D^{(k)} \mathbf{u}^{(k)} \nabla r^{(k)}))$  equations with the diffusion coefficient  $D$  given by:

$$D^{(k)} = \frac{v_t^{(k)}}{\sigma_r} \quad (141)$$

The turbulent viscosity of the dispersed air phase (d) is related to that of the continuous phase through:

$$v_t^{(d)} = \frac{v_t^{(c)}}{\sigma_f} \quad (142)$$

where  $\sigma_f$  is the turbulent Schmidt number for the interaction between the two fluids. The above described turbulence model is a modified version of the one described in [69] in which the turbulent viscosities of both fluids are allowed to be different in contrast to what is done in [69]. This is accomplished through the introduction of the  $\sigma_f$  parameter. As such, different diffusion coefficients ( $D^{(k)}$ ) are used for the different fluids. Results are compared against the experimental data reported by Seriwaza et al [61].

In the Seriwaza et al experiment [61], the Reynolds number based on superficial liquid velocity and pipe diameter is  $8 \times 10^4$ , the inlet superficial gas and liquid velocities are 0.077 and 1.36 m/s, respectively, and the inlet void fraction is  $5.36 \times 10^{-2}$  with no slip between the incoming fluids. Moreover, the bubble diameter is taken as 3 mm [69], while the fluid

properties are taken as  $\rho^{(c)}=1000 \text{ Kg/m}^3$ ,  $\rho^{(d)}=1.23 \text{ Kg/m}^3$ , and  $v_1^{(c)}=10^{-6} \text{ m}^2/\text{s}$ . The constants in the model were set to:  $C_{1a}=0.075$ ,  $\sigma_f=0.5$ ,  $\sigma_r=0.7$ , and  $C_b=0.05$ . Predicted radial profiles of the vertical liquid velocity and void fraction presented in Fig. 12 using a grid of size  $96 \times 32$  control volumes concur very well with measurements and compare favorably with numerical profiles reported by Boisson and Malin [69]. As shown, the void fraction profile indicates that gas is taken away from the pipe center towards the wall. This is caused by the lift force, which drives the bubbles towards the wall.

Having established the credibility of the physical model and numerical procedure, the next task is to assess the merits of the various algorithms for such flows. For that purpose calculations are performed using the SG, PG, and FMG strategies for all algorithms. Results are displayed in the form of (i) total mass residuals summed over both phases as a function of outer iterations (Fig. (13)), and (ii) normalized CPU time (Table 1) needed for the maximum normalized residual of all variables and for all phases to drop below  $\epsilon_s=10^{-6}$ .

As can be seen from Fig. 13, it is possible to converge the solution to the desired level with all algorithms. With the exception of PISO (Fig. 13(a)), the convergence characteristics of all algorithms (Figs. 13(b)-(g)) are very similar with PRIME, as expected, requiring the largest number of iterations. The PG method reduces, on average, the number of iterations in comparison with the SG method by about 20%. On the other hand, the FMG method results in a 50% reduction in the number of outer iterations. The use of 3 and 4 levels for both PG and FMG methods does not seem to have any effect on solution acceleration for all algorithms except PISO, for which the use of 4 levels with the FMG method increases the number of outer iterations considerably and results in a kind of oscillations (Fig. 13(a)). The convergence histories of all algorithms with the FMG method on 3 levels presented in Fig. 13(h) confirm once more the aforementioned observations.

In Table 1, the normalized CPU-times (i.e. CPU-time divided by the time needed by SIMPLE on the coarsest grid) required by the different algorithms to converge the solution to the desired level with the various methodologies are presented. For the SG method, the CPU-time on two different grids of sizes 48x16 and 96x32 C.V. are presented. As expected, the CPU effort increases for all algorithms with increasing the grid size. The PG and FMG solutions are for a grid of 96x32 C.V. using 3 (96x32, 48x16, and 24x8 C.V.) and 4 (96x32, 48x16, 24x8, and 12x4 C.V.) grid levels.

The CPU times of SIMPLE, SIMPLEC, and SIMPLEX are very close on all grids and for all methods with no clear superiority of any algorithm over the others. The performance of SIMPLEST, SIMPLEM, and PRIME is very close with SIMPLEST and PRIME being the least expensive using the SG and PG methods, respectively. With the FMG method, the PRIME algorithm is the most expensive with no clear superiority of SIMPLEST over SIMPLEM and vice versa. The performance of PISO with the SG and PG methods is very close to that of the SIMPLE algorithm. With the 4 levels FMG method however, its performance is highly unexpected necessitating higher computational effort than the SG method and may be caused by the additional explicitness introduced by the PRIME step. The use of the multi-grid method reduces the computational cost, on average, by about 45% (i.e. the FMG method is almost twice as fast as the SG method) while the use of the PG method results in a reduction of about 15%, both in comparison with the SG method. Moreover, with the FMG method, the least computational effort is obtained with SIMPLEC, which is about 0.7% less expensive than SIMPLE. Excluding PISO, the most expensive algorithm with the FMG is PRIME, which requires 27% more time than SIMPLE.

### ***Problem 2: Turbulent air-particle flow in a vertical pipe***

Here, the upward flow of a dilute gas-solid mixture in a vertical pipe is simulated. As in the previous problem, the axi-symmetric form of the gas and particulate transport equations are

employed. As reported in several studies [72-74], the effects of interfacial virtual mass and lift forces are small and may be neglected and the controlling interfacial force is drag (Harlow and Amsden [75]), which is given by:

$$\left(I_M^x\right)_D^{(c)} = -\left(I_M^x\right)_D^{(d)} = \frac{3}{8} \frac{C_D}{r_p} \rho^{(c)} r^{(d)} V_{\text{slip}} \left(u^{(d)} - u^{(c)}\right) \quad (143)$$

$$\left(I_M^y\right)_D^{(c)} = -\left(I_M^y\right)_D^{(d)} = \frac{3}{8} \frac{C_D}{r_p} \rho^{(c)} r^{(d)} V_{\text{slip}} \left(v^{(d)} - v^{(c)}\right) \quad (144)$$

where  $r_p$  represents the particle's radius,  $C_D$  the drag coefficient computed from:

$$\left\{ \begin{array}{ll} C_D = \frac{24}{Re_p} & \text{for } Re_p < 1 \\ C_D = \frac{24}{Re_p} \left(1 + 0.15 Re_p^{0.687}\right) & \text{for } 1 < Re_p < 1000 \\ C_D = 0.44 & \text{for } Re_p > 1000 \end{array} \right. \quad (145)$$

and  $Re_p$  the Reynolds number based on the particle size as defined in Eq.(134).

As before, turbulence is assumed to be a property of the continuous gas phase (c) and is predicted using a two-fluid k- $\epsilon$  model. Several extensions of the k- $\epsilon$  model for carrier-phase turbulence modulation have been proposed in the literature [50-55] and the one suggested by Chen and Wood [52], which introduces additional source terms into the turbulence transport equations, is adopted here. Thus, the turbulent viscosity is computed by solving the turbulence transport equations (Eqs. (4) and (5)) for the continuous phase with  $I_k^{(k)}$  and  $I_\epsilon^{(k)}$  evaluated using the following relations [52]:

$$I_k^{(c)} = -2\rho^{(d)} r^{(c)} r^{(d)} \frac{k^{(c)}}{\tau_p} \left(1 - e^{-0.0825 \frac{\tau_p}{\tau_\epsilon}}\right) \quad (146)$$

$$I_\epsilon^{(c)} = -2\rho^{(d)} r^{(c)} r^{(d)} \frac{\epsilon^{(c)}}{\tau_p} \quad (147)$$

where  $\tau_p$  and  $\tau_\epsilon$  are timescales characterizing the particle response and large-scale turbulent motion, respectively, and are computed from:

$$\begin{cases} \tau_p = \frac{\rho^{(d)} \mathbf{r}^{(d)}}{F_D} V_{\text{slip}} \\ \tau_e = 0.165 \frac{k^{(c)}}{\varepsilon^{(c)}} \end{cases} \quad (148)$$

with  $F_D$  being the magnitude of the inter-fluid drag force per unit volume. The turbulent eddy viscosity of the dispersed phase (d) is considered to be a function of that of the continuous phase and is computed using Eq. (142).

The above-described model is validated against the experimental results of Tsuji et al [72]. Results are replicated here for the case of an air Reynolds number, based on the pipe diameter (of value 30.5 mm), of  $3.3 \times 10^4$  and a mean air inlet velocity of 15.6 m/s using particles of diameter 200  $\mu\text{m}$  and density 1020  $\text{Kg/m}^3$ . In the computations, the mass-loading ratio at inlet is considered to be 1 with no slip between the fluids, and  $\sigma_f$  and  $\sigma_r$  are set to 5 and  $10^{10}$ , respectively (i.e. the interaction terms included for bubbly flows are neglected here). Figure 14 shows the fully developed gas and particles mean axial velocity profiles generated using a grid of size 96x40 C.V. It is evident that there is generally a very good agreement between the predicted and experimental data with the gas velocity being slightly over predicted and the particles velocity slightly under predicted. Moreover, close to the wall, the model predictions indicate that the particles have higher velocities than the gas, which is in accord with the experimental results of Tsuji et al. [72].

Having checked the correctness of the physical model and numerical procedure, the problem is solved using the SIMPLE, SIMPLEC, SIMPLEX, and SIMPLEST multi-fluid algorithms and the SG, PG, and FMG solution methods. As in the previous problem, results are displayed in the form of (i) total mass residuals summed over both phases as a function of outer iterations (Fig. (15)), and (ii) normalized CPU time (Table 2) needed for the maximum normalized residual of all variables and for all phases to drop below  $\varepsilon_s = 10^{-6}$ .

Mass residual plots presented in Fig. 15 indicate a similar convergence behavior for all four algorithms with very close number of outer iterations to achieve the desired level of convergence (i.e. within 50 outer iterations). It is hard to see any noticeable difference between the 3 and 4 levels with both the PG and FMG methods. The decrease in the number of iterations with the PG method over the SG method is smaller than the decrease obtained with bubbly flows. This lower effectiveness of the PG method is due to the following reason. In solving the problem, it is noticed that the initial guess greatly affects the convergence history and time required to converge the solution to the desired level. Except when solving on the finest mesh with the SG method, the initial guess used for the velocity field is  $u^{(c)}=u^{(d)}=1$  m/s. The use of this initial value with the SG method on the finest mesh greatly increased the CPU effort needed over the one needed when starting with an initial field of  $u^{(c)}=u^{(d)}=15.6$  m/s. To reduce cost, the latter initial guess is used. For this reason the mass residuals start from somehow a lower value than expected and the PG method appears to be less effective. The FMG method reduces the number of outer iterations by about 53% over the SG method, which indicates a good capability to deal with the added non-linearity of multiphase flows.

The normalized CPU-times presented in Table 2 reflects the above stated behaviors with the time required when using the PG method being very close (slightly higher, except with SIMPLEM it is slightly lower) to the time needed with the SG method. The FMG method reduces the cost, on average, by about 40 %. The least computational effort is accomplished with SIMPLE while SIMPLEM is the most expensive with all methods (23% more expensive than SIMPLE with the FMG on 3 levels).

### ***Problem 3: Compressible dilute air-particle flow over a flat plate***

As has been demonstrated in several studies [76-82], two-fluid flow greatly changes the main features of the boundary layer over a flat plate. Typically, three distinct regions are defined in

the two-fluid boundary layer (Fig. 16), based on the importance of the slip velocity between the two fluids: a large-slip region close to the leading edge, a moderate-slip region further down, and a small-slip zone far downstream. The characteristic scale in this two-fluid flow problem is the relaxation length  $\lambda_e$  [79], defined as:

$$\lambda_e = \frac{2}{9} \frac{\rho^{(d)} r_p^2 u_\infty}{\mu^{(c)}} \quad (149)$$

where  $u_\infty$  is the free stream velocity. The three regions are defined according to the order of magnitude of the slip parameter  $x^* = x/\lambda_e$ . In the simulation, the viscosity of the fluid is considered to be a function of temperature according to [79]:

$$\mu^{(c)} = \mu_{\text{ref}} \left( \frac{T^{(c)}}{T_{\text{ref}}} \right)^{0.6} \quad (150)$$

where the reference viscosity and temperature are  $\mu_{\text{ref}} = 1.86 \times 10^{-5} \text{ N.s/m}^2$  and  $T_{\text{ref}} = 303 \text{ }^\circ\text{K}$ .

Drag is the only retained interfacial force as it dominates the other interfacial forces. It is computed as [79]:

$$\left( \mathbf{I}_M^x \right)_D^{(c)} = - \left( \mathbf{I}_M^x \right)_D^{(d)} = \frac{9}{2} \frac{C_D}{r_p^2} \mathbf{r}^{(d)} \mu^{(c)} \left( \mathbf{u}^{(d)} - \mathbf{u}^{(c)} \right) \quad (151)$$

$$\left( \mathbf{I}_M^y \right)_D^{(c)} = - \left( \mathbf{I}_M^y \right)_D^{(d)} = \frac{9}{2} \frac{C_D}{r_p^2} \mathbf{r}^{(d)} \mu^{(c)} \left( \mathbf{v}^{(d)} - \mathbf{v}^{(c)} \right) \quad (152)$$

where the drag coefficient is given by:

$$C_D = \frac{1}{50} \text{Re}_p + \frac{7}{6} \text{Re}_p^{0.15} \quad (153)$$

In the energy equation, heat transfer due to radiation is neglected and only convective heat transfer around an isolated particle is considered. Under such conditions, the interfacial terms in the gas (c) and particles (d) energy equations reduce to [79]:

$$\mathbf{I}_E^{(c)} = \mathbf{Q}_{g-p} + \mathbf{F}_{g-p} \cdot \mathbf{u}^{(d)} \quad (154)$$

$$\mathbf{I}_E^{(d)} = -\mathbf{Q}_{g-p} \quad (155)$$

where:

$$\mathbf{F}_{g-p} = \left( \mathbf{I}_M^x \right)_D^{(c)} \mathbf{i} + \left( \mathbf{I}_M^y \right)_D^{(c)} \mathbf{j} \quad (156)$$

$$\text{Nu} = 2.0 + 0.6 \text{Re}_p^{\frac{1}{2}} \left( \text{Pr}^{(c)} \right)^{\frac{1}{3}} \quad (157)$$

$$Q_{g-p} = \frac{3}{2} \frac{r^{(d)} \lambda^{(c)} \text{Nu}}{r_p^2} \left( T^{(d)} - T^{(c)} \right) \quad (158)$$

In the above equations, Nu is the Nusselt number,  $\text{Pr}^{(c)}$  the gas Prandtl number,  $\lambda^{(c)}$  the gas thermal conductivity, T the temperature, and other parameters are as defined earlier.

In the simulation, the particle diameter, particle Reynolds number, material density, Prandtl number, and mass load ratio are set to: 10  $\mu\text{m}$ , 10, 1766  $\text{kg/m}^3$ , 0.75, and 1 respectively. The wall boundary is treated as a no-slip boundary for the gas phase (i.e. both components of the gas velocity are set to zero), and as a slip boundary condition for the particles phase (i.e. the normal fluxes are set to zero). In order to bring all quantities to the same order of magnitude, results are displayed using the following dimensionless variables:

$$x^* = \frac{x}{\lambda_e}, y^* = \frac{y}{\lambda_e} \sqrt{\text{Re}}, u^* = \frac{u}{u_\infty}, v^* = \frac{v}{u_\infty} \sqrt{\text{Re}} \quad \text{Re} = \frac{\rho u \lambda_e}{\mu} \quad (159)$$

Figure 17 shows the results for the steady flow obtained on a rectangular domain with a mesh of density 104x48 C.V. stretched in the y-direction. The figure depicts the development of gas and particles velocity profiles within the three regions mentioned earlier. In the near leading edge area ( $x^*=0.1$ ), the gas velocity is adjusted at the wall to obtain the no-slip condition as for the case of a pure gas boundary layer. The particles have no time to adjust to the local gas motion and there is a large velocity slip between the fluids. In the transition region ( $x^*=1$ ), significant changes in the flow properties take place. The interaction between the fluids cause the particles to slow down while the gas accelerates. In the far downstream region ( $x^*=5$ ), the particles have ample time to adjust to the state of the gas motion, the slip is very small, and the solution tends to equilibrium. These results are in excellent agreement with numerical solutions reported by Thevand et al. [82] (Fig. 17).

As in test 1, the problem is solved using all multi-fluid algorithms and the SG, PG, and FMG solution methods. Results are displayed in the form of total mass residuals (Fig. 18) and normalized CPU time (Table 3) with  $\epsilon_s=10^{-6}$ .

Plots presented in Fig. 18 indicate that it is possible to converge the solution to the desired level with all algorithms. As depicted in figure 18(a), PISO requires the least number of outer iterations. This, however, is not associated with the lowest computational effort due to the higher cost per iteration in comparison with other algorithms. The convergence characteristics of SIMPLE (Fig. 18(b)), SIMPLEC (Fig. 18(c)), SIMPLEMENT (Fig. 18(d)), and SIMPLEX (Fig. 18(g)) are very similar, requiring nearly the same number of outer iterations with the SG, PG, and FMG methods. The number of iterations required by PRIME (Fig. 18(f)) with the SG method is higher than SIMPLEST (Fig. 18(e)). With the FMG method however, the performance of the two algorithms is very close with that of PRIME being slightly better. In general, the use of the PG method reduces the number of outer iterations, as compared to the SG method, by over 40% with all algorithms whereas the use of the FMG method reduces it by over 64%. Fig. 18(h) indicates that when using the FMG method on 3 levels PISO requires the lowest number of iterations followed by SIMPLEMENT, SIMPLEC, SIMPLEX, SIMPLE, PRIME, and SIMPLEST. With the number of iterations needed by SIMPLEST and PRIME being very close and nearly double the number needed by SIMPLE. It should be clarified that the displayed numbers of iterations represent those needed for the mass residuals to be reduced to the desired level. The CPU-times however represent the computational effort needed to reduce the maximum residuals to the desired level. In some cases, even though the mass residuals may become below the desired value, other residuals could still be above that value. This is why, for example, the CPU time needed by SIMPLE is lower than that needed by SIMPLEC (Table 3) even though Fig. 18(h) indicates that the number of iterations needed by SIMPLEC to reduce the mass residuals to below  $\epsilon_s$  is lower.

In Table 3, the normalized CPU-times required by the different algorithms to solve the problem with the various methodologies are presented. For the SG method, the CPU-time on two different grids of sizes 52x24 and 104x48 C.V. are presented. As expected, the CPU effort increases for all algorithms with increasing the grid size. The PG and FMG solutions are for a grid of 104x48 C.V. using 3 (104x48, 52x24, and 26x12 C.V.) and 4 (104x48, 52x24, 26x12, and 13x6 C.V.) grid levels. The SIMPLE algorithm appears to be the most efficient requiring the least computational effort on all grids and with all methods. The performance of SIMPLEC and SIMPLEX is the closest to SIMPLE especially with the multi-grid method. SIMPLEST and PRIME are the most expensive to use with SIMPLEST performing better with the SG and PG methods (PRIME requires 170% more time than SIMPLE on the dense grid with the SG method while SIMPLEST requires 54% additional time). PISO is more expensive than SIMPLE on all grids and with all methods (it requires 15% more time than SIMPLE on the dense grid with the SG method and 56% more time with the FMG on 3 levels).

#### ***Problem 4: Inviscid transonic dusty flow in a converging-diverging nozzle***

The last test considered deals with the prediction of supersonic dilute air-particle flow in an axi-symmetric converging-diverging rocket nozzle. Several researchers have analyzed the problem and data is available for comparison [83-92]. In most of the reported studies, a shorter diverging section, in comparison with the one considered here, has been used when predicting the two-fluid flow. Two-fluid flow results for the long configuration have only been reported by Chang et.al. [87]. The flow is assumed to be inviscid and the single-fluid flow results are used as an initial guess for solving the two-fluid flow problem. The physical configuration (Fig. 19) is the one described in [87]. The viscosity of the fluid varies with the temperature according to Sutherland's law for air:

$$\mu^{(c)} = 1.458 \times 10^{-6} \frac{T^{(c)} \sqrt{T^{(c)}}}{T^{(c)} + 110.4} \quad (160)$$

The coupling between gas and particle phases is through the interfacial momentum and energy terms. The force exerted on a single particle moving through a gas [88] is given as:

$$f_x = 6\pi r_p f_D \mu^{(c)} (u^{(d)} - u^{(c)}) \quad (161)$$

$$f_y = 6\pi r_p f_D \mu^{(c)} (v^{(d)} - v^{(c)}) \quad (162)$$

so that for  $N$  particles in a unit volume the effective drag force is

$$(I_M^x)_D^{(c)} = -(I_M^x)_D^{(d)} = \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (u^{(d)} - u^{(c)}) \quad (163)$$

$$(I_M^y)_D^{(c)} = -(I_M^y)_D^{(d)} = \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (v^{(d)} - v^{(c)}) \quad (164)$$

where  $f_D$  is the ratio of the drag coefficient  $C_D$  to the stokes drag  $C_{D0}=24/Re_p$  and is given by [87]

$$f_D = 1 + 0.15 Re_p^{0.687} + \frac{0.0175 Re_p}{1 + 4.25 \times 10^4 Re_p^{-1.16}} \quad Re_p < 3 \times 10^5 \quad (165)$$

The heat transferred from gas to particle phase per unit volume is given as [88]

$$Q_{g-p} = \frac{3}{2} \frac{r^{(d)}}{r_p} \lambda^{(c)} Nu (T^{(d)} - T^{(c)}) \quad (166)$$

Where  $\lambda^{(c)}$  is the thermal conductivity of the gas and  $Nu$  the Nusselt number, which is written as [88]

$$Nu = 2 + 0.459 Re_p^{0.55} Pr_c^{0.33} \quad (167)$$

The gas-particle inter-fluid energy term is given by

$$I_E^{(c)} = \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (u^{(d)} - u^{(c)}) u_d + \frac{9}{2} \frac{r^{(d)}}{r_p^2} f_D \mu^{(c)} (v^{(d)} - v^{(c)}) v_d + \frac{3}{2} \frac{r^{(d)}}{r_p} \lambda^{(c)} Nu (T^{(d)} - T^{(c)}) \quad (168)$$

$$I_E^{(d)} = \frac{3}{2} \frac{r^{(d)}}{r_p} \lambda^{(c)} Nu (T^{(c)} - T^{(d)}) \quad (169)$$

where the first two terms on the right-hand side of equation (168) represent the energy exchange due to momentum transfer.

The physical quantities employed are similar to those used in [87]. The gas stagnation temperature and pressure at inlet to the nozzle are 555 °K and  $10.34 \times 10^5 \text{ N/m}^2$ , respectively. The specific heat for the gas and particles are  $1.07 \times 10^3 \text{ J/Kg}^\circ\text{K}$  and  $1.38 \times 10^3 \text{ J/Kg}^\circ\text{K}$ , respectively, and the particle density is  $4004.62 \text{ kg/m}^3$ . With a zero inflow velocity angle, the fluid is accelerated from subsonic to supersonic speed in the nozzle. The inlet velocity and temperature of the particles are presumed to be the same as those of the gas phase. Results for two particle sizes of radii 1 and  $10 \text{ }\mu\text{m}$  with the same mass fraction  $\phi=0.3$  are presented using a grid of size  $188 \times 80 \text{ C.V.}$  Figures 20(a) and 20(b) show the particle volume fraction contours while Figures 20(c) and 20(d) display the velocity distribution. For the flow with particles of radius  $1 \text{ }\mu\text{m}$ , a sharp change in particle density is obtained near the upper wall downstream of the throat, and the particle density decreases to a small value. With the large particle flow ( $10 \text{ }\mu\text{m}$ ), however, a much larger particle-free zone appears due to the inability of the heavier particles to turn around the throat corner. These findings are in excellent agreement with published results reported in [87] and others using different methodologies. A quantitative comparison of current predictions with published experimental and numerical data is presented in Fig. 21 through gas Mach number distributions along the wall (Fig. 21(a)) and centerline (Fig. 21(b)) of the nozzle for the one-fluid and two-fluid flow situations. As can be seen, the one-fluid flow predictions fall on top of experimental data reported in [90-92]. Since the nozzle contour has a rapid contraction followed by a throat with a small radius of curvature, the flow near the throat wall is overturned and inclined to the downstream wall. A weak shock is thus formed to turn the flow parallel to the wall. This results in a sudden drop in the Mach number value and as depicted in Fig. 21(b), this sudden drop is correctly envisaged by the solution algorithm with the value after the shock being slightly over predicted.

Due to the unavailability of experimental data, two-fluid flow predictions are compared against the numerical results reported in [87]. As displayed in Figs. 21(a) and 21(b), both solutions are in good agreement with each other indicating once more the correctness of the calculation procedures. The lower gas Mach number values in the two-fluid flow is caused by the heavier particles ( $\rho^{(d)} \gg \rho^{(c)}$ ), which reduce the gas velocity. Moreover, owing to the particle-free zone, the Mach number difference between the one- and two-fluid flows along the wall is smaller than that at the centerline.

To compare the relative performance of the multi-fluid algorithms, the problem is solved via the PG method using the SIMPLE, SIMPLEC, SIMPLEM, and SIMPLEX algorithms over three different grids of sizes 47x20, 94x40, and 188x80 C.V. for a particle radius of size 1  $\mu\text{m}$ . As before, results are displayed in the form of total mass residuals (Fig. 22) and normalized CPU times (Table 4) with  $\varepsilon_s$  set to  $10^{-5}$ . As shown in Fig. 22, all algorithms require almost the same number of iterations with the exception of SIMPLE on the 94x40 grid, which requires a larger number of iterations than on the finest mesh. Excluding that case, the convergence histories of all algorithms are nearly identical. In terms of computational effort, the normalized CPU-times presented in Table 4 indicate a close performance of the various algorithms with SIMPLE being the most efficient (7% less expensive than SIMPLEC) and SIMPLEM the most expensive (43% more expensive than SIMPLE) on the finest mesh. On the other hand, SIMPLEX is 11% more expensive than SIMPLE.

## Closing Remarks

The implementation of seven GCBA multi-fluid algorithms for the simulation of multi-fluid flow at all speeds was accomplished. The algorithms were embedded within a non-linear full multi-grid strategy. A two-fluid k- $\varepsilon$  model and several inter-phase models were also

employed. Solving a variety of one- and two-dimensional two-phase flow problems assessed the performance and accuracy of these algorithms. For each test problem, solutions were generated on a number of grid systems using the single grid method (SG), the prolongation grid method (PG), and the full non-linear multi-grid method (FMG). Results obtained demonstrated the capability of all algorithms to deal with multi-fluid flow situations and to predict multi-fluid flow at all speeds, and the ability of the FMG method to tackle the added non-linearity of laminar and turbulent multi-fluid flows. The convergence history plots and CPU-times presented, indicated similar performances for SIMPLE, SIMPLEC, and SIMPLEX. The PISO, SIMPLEM, and SIMPLEST algorithms were in general more expensive than SIMPLE. In general, the PRIME algorithm was the most expensive to use. The PG and FMG methods accelerated the convergence rate for all algorithms. The FMG method was found to be more efficient.

## **Acknowledgments**

The European Office of Aerospace Research and Development (EOARD) (SPC01-4005) gratefully supported this work.

## References

1. Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," Math. Comp., vol. 31, pp. 333-390, 1977.
2. Rhie, C.M., | "A Pressure Based Navier-Stokes Solver Using The Multigrid Method" AIAA paper 86-0207, 1986.
3. Wesseling, P., "An introduction to Multigrid Methods," John Wiley & Sons Ltd., Baffins Lane, Chichester, West Sussex PO19 1UD, England, 1995.
4. Dick, E., "Multigrid Formulation of polynomial flux-difference splitting for steady Euler Equations," Journal of Computational Physics, vol. 91, pp. 161-173, 1990.
5. Kershaw, D., "The Incomplete Cholesky-Conjugate Gradient Method for The Iterative Solution of Systems of Linear Equations," Journal of Computational Physics, vol. 26, pp. 43-65, 1978.
6. Stone, H.L., "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations," SIAM J. Num. Anal., vol. 5, No. 3, pp. 530-558, 1968.
7. Venkatakrishnan V., Mavriplis D.J. "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids", AIAA paper 95-1705, 1995
8. Venkatakrishnan V, "Perspective on Unstructured Grid Solvers", AIAA Journal, vol. 34, No. 3, pp, 533-547, 1996.
9. Whitaker D.L., "Three Dimensional Unstructured Grid Euler Computations Using a Fully-Implicit, Upwind Method", AIAA paper 93-3357, 1993
10. Oan D. Cheng J.C, "Upwind Finite Volume Navier-Stokes Computations on Unstructured Triangular Meshes", AIAA J., vol 31 No. 9, pp. 1618-1625, 1993
11. Mitchell C.R. "Improved Reconstructuion Schemes for the Navier-Stokes Equations on Unstructured Grids, AIAA paper 94-0642, 1994.
12. Barth T.J, Jespersen D.C., "The Design and Application of Upwind Schemes on Unstructured Meshes", AIAA paper 89-0366, Jan. 1989
13. Gaskell, P.H. and Lau, A.K.C., "Curvature compensated Convective Transport: SMART, a new boundedness preserving transport algorithm," Int. J. Num. Meth. Fluids, vol. 8, pp. 617-641, 1988.
14. Leonard, B.P., "Locally Modified Quick Scheme for Highly Convective 2-D and 3-D Flows," Taylor, C. and Morgan, K. (eds.), Numerical Methods in Laminar and Turbulent Flows, Pineridge Press, Swansea, U.K., vol. 15, pp. 35-47, 1987.
15. Darwish, M.S. and Moukalled, F., "Normalized Variable and Space Formulation Methodology For High-Resolution Schemes," Numerical Heat Transfer, Part B, vol. 26, pp. 79-96, 1994.

16. Moukalled, F. and Darwish, M.S., "A New Family of Streamline-Based Very High Resolution Schemes" *Numerical Heat Transfer*, vol. 32 No 3, pp. 299-320, 1997.
17. Darwish, M. and Moukalled, F., "An Efficient Very High-Resolution scheme Based on an Adaptive-Scheme Strategy," *Numerical Heat Transfer, Part B*, vol. 34, pp. 191-213, 1998.
18. Moukalled, F. and Darwish, M., "New Family of Adaptive Very High Resolution Schemes," *Numerical Heat Transfer, Part B*, vol. 34, pp. 215-239, 1998.
19. Marchi, C.H. and Maliska, C.R., "A Non-orthogonal Finite-Volume Methods for the Solution of All Speed Flows Using Co-Located Variables," *Numerical Heat Transfer, Part B*, vol. 26, pp. 293-311, 1994.
20. Demirdzic, I., Lilek, Z., and Peric, M., "A Collocated Finite Volume Method For Predicting Flows at All Speeds," *International Journal for Numerical Methods in Fluids*, vol. 16, pp. 1029-1050, 1993.
21. Lien, F.S. and Leschziner, M.A., "A General Non-Orthogonal Collocated Finite Volume Algorithm for Turbulent Flow at All Speeds Incorporating Second-Moment Turbulence-Transport Closure, Part 1: Computational Implementation," *Computer Methods in Applied Mechanics and Engineering*, vol. 114, pp. 123-148, 1994.
22. Politis, E.S. and Giannakoglou, K.C., "A Pressure-Based Algorithm for High-Speed Turbomachinery Flows," *International Journal for Numerical Methods in Fluids*, vol. 25, pp. 63-80, 1997.
23. Chen, K.H. and Pletcher, R.H., "Primitive Variable, Strongly Implicit Calculation Procedure for Viscous Flows at All Speeds," *AIAA Journal*, vol. 29, no. 8, pp. 1241-1249, 1991.
24. Darbandi, M. and Shneider, G.E., "Momentum Variable Procedure for Solving Compressible and Incompressible Flows," *AIAA Journal*, Vol. 35, No. 12, pp. 1801-1805, 1997.
25. Darbandi, M. and Schneider, G.E., "Use of a Flow Analogy in Solving Compressible and Incompressible Flows," *AIAA Paper 97-2359*, Jan. 1997.
26. Karki, K.C., "A Calculation Procedure for Viscous Flows at All Speeds in Complex Geometries," Ph.D. Thesis, University of Minnesota, June 1986.
27. Moukalled, F. and Darwish, M., "A High-Resolution Pressure-Based Algorithm for Fluid Flow at All Speeds," *Journal of Computational Physics*, vol. 168, no. 1, pp. 101-133, 2001.
28. Patankar, S.V. and Spalding, D.B., "A Calculation Procedure for Heat Mass and Momentum Transfer in Three Dimensional Parabolic Flows", *Int. J. Heat & Mass Trans.*, vol. 15, pp. 1787, 1972
29. Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, N.Y., 1981.

30. Issa, R.I., "Solution of the Implicit Discretized Fluid Flow Equations by Operator Splitting," Mechanical Engineering Report, FS/82/15, Imperial College, London, 1982.
31. Van Doormaal, J. P. and Raithby, G. D., "Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows' Numerical Heat Transfer, vol. 7, pp. 147-163, 1984.
32. Van Doormaal, J. P. and Raithby, G. D., "An Evaluation of the Segregated Approach for Predicting Incompressible Fluid Flows," ASME Paper 85-HT-9, Presented at the National Heat Transfer Conference, Denver, Colorado, August 4-7, 1985.
33. Acharya, S. and Moukalled, F., "Improvements to Incompressible Flow Calculation on a Non-Staggered Curvilinear Grid," Numerical Heat Transfer, Part B, vol. 15, pp. 131-152, 1989.
34. Maliska, C.R. and Raithby, G.D., "Calculating 3-D fluid Flows Using non-orthogonal Grid," Proc. Third Int. Conf. on Numerical Methods in Laminar and Turbulent Flows, Seattle, pp. 656-666, 1983.
35. Moukalled, F. and Darwish, M., "A Unified Formulation of the Segregated Class of Algorithms for Fluid Flow at All Speeds," Numerical Heat Transfer, Part B: Fundamentals, vol. 37, No 1, pp. 103-139, 2000.
36. Darwish, M., Moukalled, F., and, Sekar, B., "A Unified Formulation of the Segregated Class of Algorithms for Multi-Fluid Flow at All Speeds," Numerical Heat Transfer, Part B: Fundamentals, vol. 40, no. 2, pp. 99-137, 2001.
37. Spalding D.B., "The Calculation of Free-Convection Phenomena in Gas-Liquid Mixtures" Report HTS/76/11 Mech. Eng. Imperial College, London, 1976
38. Spalding, D.B., "Numerical Computation of Multi-Phase Fluid Flow and Heat Transfer", in Recent Advances in Numerical Methods in Fluid eds. Taylor C., Morgan K., pp. 139-167, vol. 1, 1980
39. Spalding, D.B., "A General Purpose Computer Program for Multi-Dimensional, One and Two Phase Flow", Report HTS/81/1 Mech. Eng. Imperial College, London, 1981
40. Rivard, W.W. and Torrey, M.D., "KFIX: A Program for Transient Two Dimensional Two Fluid Flow", Report LA-NUREG-6623, 1978
41. Amsden, A.A., Harlow F.H., "KACHINA: An Eulerian Computer Program for Multifield Flows", Report LA-NUREG-5680, 1975
42. Amsden, A.A., Harlow F.H., "KTIFA Two-Fluid Computer Program for Down comer Flow Dynamics", Report LA-NUREG-6994, 1977
43. Moukalled, F., "A Pressure-Based Finite Volume Method with Full Multi-Grid Acceleration for Multi-Fluid Flow at All Speeds," EOARD Report, 2001.
44. Moukalled, F. and Darwish, M., "A Comparative Assessment of the

- Performance of Mass Conservation Based Algorithms for Incompressible Multiphase Flows,” Numerical Heat Transfer, Part B: Fundamentals, (in print).
45. Moukalled, F., Darwish, M., and Sekar, B.” A High Resolution Pressure-Based Algorithm for Multi-Phase Flow at all Speeds,” Journal of Computational Physics, (in review).
  46. Darwish, M., Moukalled, F., and Sekar, B.” A Robust Multi-Grid Pressure-Based Algorithm for Multifluid Flow at All Speeds,” International Journal for Numerical Methods in Fluids, (in review).
  47. Baldwin, B.S. and Lomax, H.” Thin Layer Approximation and algebraic model for separated turbulent flows,” AIAA Paper 78-257, 1978.
  48. Sotiropoulos, F. and Patel, V.C.” Application of Reynolds-stress Transport Models to Stern and Wake Flow,” J. Ship Res., vol. 39, p. 263, 1995.
  49. Cokljat, D., Ivanov, V.A., Srasola, F.J., and Vasquez, S.A.” Multiphase K-Epsilon Models for Unstructured Meshes,” ASME 2000 Fluids Engineering Division Summer Meeting, June 11-15, 2000, Boston Massachusetts, USA.
  50. Pourahmadi, F. and Humphrey, J.A.C.” Modeling solid-fluid turbulent flows with application to predicting erosive wear,” Int. J. Phys. Chem. Hydro, vol. 4, pp. 191-219, 1983.
  51. Elghobashi, S.E. and Abou-Arab, T.W.” A two-equation turbulence model for two-phase flows,” Phys. Fluids, vo. 26, no. 4, pp. 931-938, 1983.
  52. Chen, C.P. and Wood, P.E.” Turbulence closure modeling of the dilute gas-particle axisymmetric jet,” A.I.C.H.E Journal, vol. 32, no. 1, pp. 163-166, 1986.
  53. Lopez de Bertodano, M., Lee, S.J., Lahey, R.T. Jr., and Drew, D.A.” The prediction of two-phase turbulence and phase distribution phenomena using a Reynolds stress model,” ASME Journal of Fluids Engineering, vo. 112, pp. 107-113, 1990.
  54. Lopez de Bertodano, M., Lahey, R.T. Jr., and Jones, O.C.” Development of a  $k-\epsilon$  model for bubbly two-phase flow,” ASME Journal of Fluids Engineering, vo. 116, pp. 128-134, 1994.
  55. Lopez de Bertodano, M., Lahey, R.T. Jr., and Jones, O.C.” Phase distribution in bubbly two-phase flow in vertical ducts,” Int. J. Multiphase flow, vol. 20, no. 5, pp. 805-818, 1994.
  56. Ferziger, J.H. and Peric, M., Computational Methods for Fluid Dynamics, Springer-Verlag, Berlin Heidelberg, 1996.
  57. Thompson, C.P. and Lezeau, P.” Application of the Full Approximation Storage Method to the Numerical Simulation of Two-Dimensional Steady Incompressible Viscous Multifluid Flows,” International Journal for Numerical Methods in Fluids, vol. 28, #8, pp. 1217-1239, 1998.
  58. Cornelius, C., Volgmann, W., and Stoff, H.” Calculation of Three-Dimensional Turbulent Flow with a Finite Volume Multigrid Method,” International

- Journal for Numerical Methods in Fluids, vol. 31, pp. 703-720, 1999.
59. Morsi, S.A. and Alexander, A.J., "An investigation of Particle Trajectories in Two-Phase Flow System," *Journal of Fluid Mechanics*, vol. 55, part 2, pp. 193-208, 1972.
  60. Baghdadi, A.H.A., "Numerical Modelling of Two-Phase Flow With Inter-Phase Slip," Ph.D. Thesis, Imperial College, University of London, 1979.
  61. Serizawa, A., Kataoka, I., and Michiyoshi, I., "Phase Distribution in Bubbly Flow," Data set No. 24, *Proceedings of the Second International Workshop on Two-Phase Flow Fundamentals*, Rensselaer Polytechnic Institute, Troy, NY, 1986.
  62. Wang, S.k., Lee, S-j, Jones, Jr., O.C., and Lahey, Jr., R.T., "3-D Turbulence structure and Phase Distribution Measurements in Bubbly Two-Phase Flows," *International Journal of Multiphase Flow*, vol. 13, No. 3, 1987.
  63. Antal, S.P., Lahey, R.T., and Flaherty, J.E., "Analysis of Phase Distribution in Fully Developed Laminar Bubbly Two-Phase Flows," *International Journal of Multiphase Flow*, vol. 17, No. 5, pp. 635-652, 1991.
  64. Sato, Y., sadatomi, M, and Sekoguchi, K., "Momentum and Heat Transfer in Two-Phase Bubble flow," *International Journal of Multiphase Flow*, I. Theory, pp. 167-177, II. A Comparison Between Experiment and Theoretical Calculations, pp. 179-190, 1981.
  65. Lopez de Bertodano, M., Lee, S-J., Lahey, Jr., R.T., and Drew, D.A., "The Prediction of Two-Phase Turbulence and Phase Distribution Phenomena Using a Reynolds Stress Model," *Journal of Fluids Engineering*, vo. 112, pp. 107-113, 1990.
  66. Lopez de Bertodano, M., Lahey, Jr., R.T., and Jones, O.C., "Development of a k- $\epsilon$  Model for Bubbly Two-Phase Flow," *Journal of Fluids Engineering*, vo. 116, pp. 128-134, 1994.
  67. Lopez de Bertodano, M., Lahey, Jr., R.T., and Jones, O.C., "Phase Distribution in Bubbly Two-Phase Flow in Vertical Ducts," *International Journal of Multiphase Flow*, vol. 20, No. 5, pp. 805-818, 1994.
  68. Nakoryakov, V.E., Kashinsky, O.N., Randin, V.V., and Timkin, L.S., "Gas-Liquid Bubbly Flow in Vertical Pipes," *Journal of Fluids Engineering*, vo. 118, pp. 377-382, 1996.
  69. Boisson, N. and Malin, M.R., "Numerical Prediction of Two-Phase Flow in Bubble Columns," *International Journal for Numerical Methods in Fluids*, vol. 23, pp. 1289-1310, 1996.
  70. Lahey, R.T., Lopez de Bertodano, M., and Jones, O.C., "Phase Distribution in Complex Geometry Ducts," *Nuclear Engineering Design*, vol. 141, p. 177, 1993.
  71. Drew, D.A. and Lahey, Jr., T.J., "The Virtual Mass and Lift Force on a Sphere in Rotating and Straining Inviscid Flow," *International Journal of Multiphase*

- Flow, vol. 13, No. 1, p. 113, 1987.
72. Tsuji, Y., Morikawa, Y., and Shiomi, H., "LDV Measurements of an Air-Solid Two-Phase Flow in a Vertical Pipe," *Journal of Fluid Mechanics*, vo. 139, pp. 417-434, 1984.
  73. Adeniji-Fashola, A. and Chen, C.P., "Modeling of Confined Turbulent Fluid-Particle Flows Using Eulerian and Lagrangian Schemes," *International Journal of Heat and Mass transfer*, vol.33, pp. 691-701, 1990.
  74. Naik, S. and Bryden, I.G., "Prediction of Turbulent Gas-Solids Flow in Curved Ducts Using The Eulerian-Lagrangian Method," *International Journal for Numerical Methods in Fluids*, vol. 31, pp. 579-600, 1999.
  75. Harlow, F.H. and Amsden, A.A., "Numerical Calculation of Multiphase Fluid Flow," *Journal of Computational Physics*, vol. 17, pp. 19-52, 1975.
  76. Osipov, A.N., "Structure of the Laminar Boundary Layer of a Disperse Medium on a Flat Plate," *Fluid Dyn.*, vol. 15, pp. 512-517, 1980.
  77. Prabha, S. and Jain, A.C., "On the Use of Compatibility Conditions in the Solution of Gas Particulate Boundary Layer Equations," *Appl. Sci. res.*, vol. 36, pp. 81-91, 1980.
  78. Sgleton, R.E., "The Incompressible Gas Solid Particle Flows Over a Semi-Infinite Flat Plate," *Z. Angew Math. Phys.*, vol. 19, p. 545, 1965.
  79. Wang, B.Y., and Glass, I.I., "Compressible Laminar Boundary Layer Flows of a Dusty Gas Over a Semi-Infinite Flat Plate," *Journal of Fluid Mechanics*, vol. 186, pp. 223-241, 1988.
  80. Soo, S.L., "Boundary Layer Motion of a Gas-Solids Suspension," *Proc. Symp. Interaction Between Fluids and Particles*, Inst. Chem. Eng., pp. 50-63, 1962.
  81. Chamkha, A.J., and Peddieson, J.J.R., "Boundary Layer Flow of a Particulate Suspension Past a Flat Plate," *International Journal of Multiphase Flow*, vol. 17, pp. 805-808, 1991.
  82. Thevand, N., Daniel, E., and Loraud, J.C., "On high-Resolution Schemes for Solving Unsteady Compressible Two-Phase Dilute Viscous Flows," *International Journal of Numerical Methods in Fluids*, vol. 31, pp. 681-702, 1999.
  83. Chang, I.S., "One and Two-Phase Nozzle Flows," *AIAA J.*, vol. 18, pp. 1455-1461, 1980.
  84. Ishii, R. and Kawasaki, K., "Limiting Particle Streamline in the flow of a Gas-Particle Mixture Through an Axially Symmetric Nozzle," *Phys. Fluids*, vol. 25, No. 6, pp. 959-966, 1982.
  85. Ishii, R., Umeda, Y., and Kawasaki, K., "Nozzle Flows of Gas-Particle Mixtures," *Phys. Fluids*, vol. 30, No. 3, pp. 752-760, 1987.
  86. Hwang, C.J. and Chang, G.C., "Numerical Study of Gas-Particle Flow in a Solid Rocket Nozzle," *AIAA Journal*, vol. 26, No. 6, pp. 682-689, 1988.

87. Chang, H.T., Houn, L.W., and Chien, L.E., "Application of Flux-Vector-Splitting Scheme to a Dilute Gas-Particle JPL Nozzle Flow," *International Journal for Numerical Methods in Fluids*, vol. 22, pp. 921-935, 1996.
88. Mehta, R.C. and Jayachandran, T., "A Fast Algorithm to Solve Viscous Two-Phase Flow in an Axisymmetric Rocket Nozzle," *International Journal for Numerical Methods in Fluids*, vol. 26, pp. 501-517, 1998.
89. Igra, O., Elperin, I., and Ben-Dor, G., "Dusty Gas Flow in a Converging-Diverging Nozzle," *Journal of Fluids Engineering*, vol. 121, pp. 908-913, 1999.
90. Back, L.H. and Cuffel, R.F., "Detection of Oblique Shocks in a Conical Nozzle with a Circular-arc Throat," *AIAA Journal*, vol. 4, pp. 2219-2221, 1966.
91. Back, L.H. Massier, P.F., and Cuffel, R.F., "Flow Phenomena and Convective Heat Transfer in a Conical Supersonic Nozzle," *Journal of Spacecraft*, vol. 4, pp. 1040-1047, 1967.
92. Cuffel, R.F., Back, L.H., and Massier, P.F., "Transonic Flowfield in a Supersonic Nozzle with Small Throat Radius of Curvature," *AIAA Journal*, vol. 7, pp. 1364-1366, 1969.

## Figure Captions

Fig. 1 (a) Control volume, (b) the prolongation only, and (c) FMG strategies, and (d) Physical domain for the gas-particle transport problem.

Fig. 2 (a) Comparison between the analytical and numerical particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dilute gas-solid flow problem.

Fig. 3 (a) gas and particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense gas-solid flow problem.

Fig. 4 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dilute bubbly flow problem.

Fig. 5 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dense bubbly flow problem.

Fig. 6 Normalized CPU-times for the horizontal (a) dilute gas-solid, (b) dense gas-solid, (c) dilute bubbly, and (d) dense bubbly flow problem.

Fig. 7 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dilute gas-solid flow problem.

Fig. 8 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense gas-solid flow problem.

Fig. 9 (a) gas and particle velocity distributions; and (b)-(h) convergence histories on the different grid systems for the vertical dilute bubbly flow problem.

Fig. 10 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense bubbly flow problem.

Fig. 11 Normalized CPU-times for the vertical (a) dilute gas-solid, (b) dense gas-solid,

(c) dilute bubbly, and (d) dense bubbly flow problem.

Fig. 12 Comparison of fully developed liquid velocity and void fraction profiles for turbulent bubbly upward bubbly flow in a pipe.

Fig. 13 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for turbulent upward bubbly flow in a pipe.

Fig. 14 Comparison of fully developed gas and particle velocity profiles for turbulent air-particle flow in a pipe.

Fig. 15 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEST, and (d) SIMPLEX algorithms using the SG, PG, and FMG methods on the finest mesh for turbulent air-particle flow in a pipe.

Fig. 16 The three different regions within the boundary layer of dusty flow over a flat plate.

Fig. 17 Comparison of fully developed gas and particle velocity profiles inside the boundary layer at different axial locations for dilute two-phase flow over a flat plate.

Fig. 18 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for dusty gas flow over a flat plate.

Fig. 19 Physical domain for the dusty gas flow in a converging-diverging nozzle.

Fig. 20 (a,b) Volume Fraction contours and (c,d) particle velocity vectors for dusty gas flow in a converging-diverging nozzle.

Fig. 21 Comparison of one-phase and two-phase gas Mach number distributions along the (a) wall and (b) centerline of the dusty flow in a converging-diverging nozzle problem.

Fig. 22 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEM, and (d) SIMPLEX algorithms using the SG method for dusty gas flow in a converging-diverging nozzle.

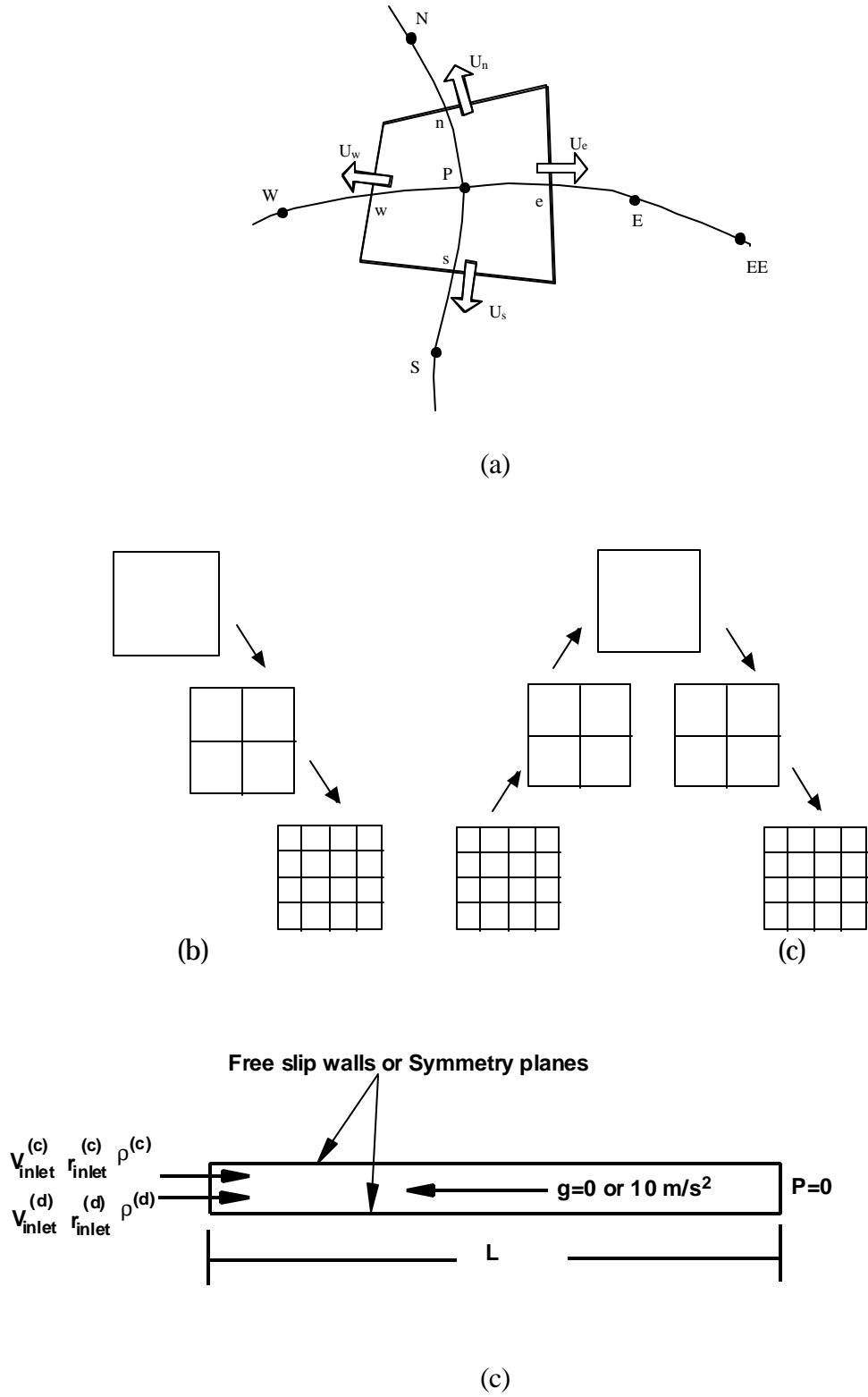
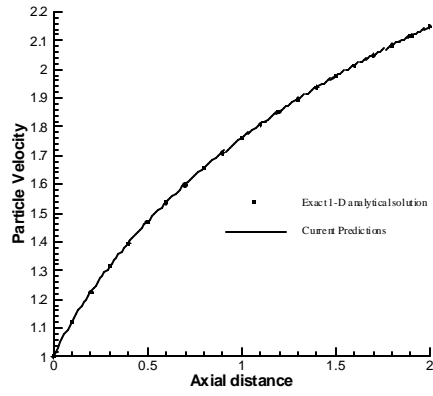
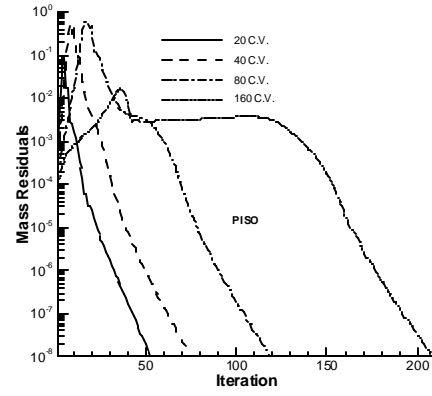


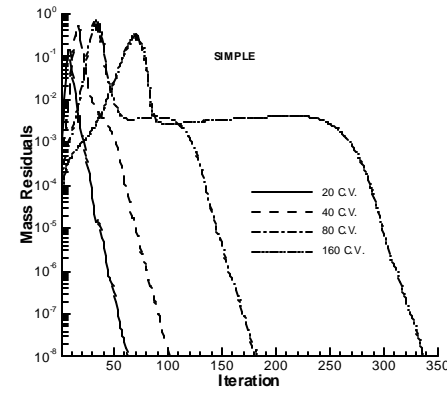
Fig. 1 (a) Control volume, (b) the prolongation only, and (c) FMG strategies, and (d) Physical domain for the gas-particle transport problem.



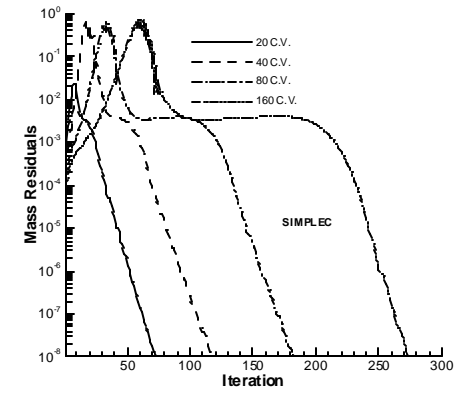
(a)



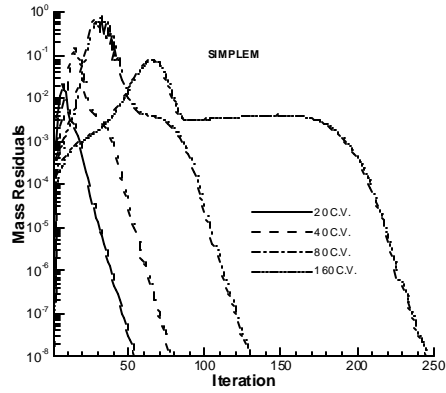
(b)



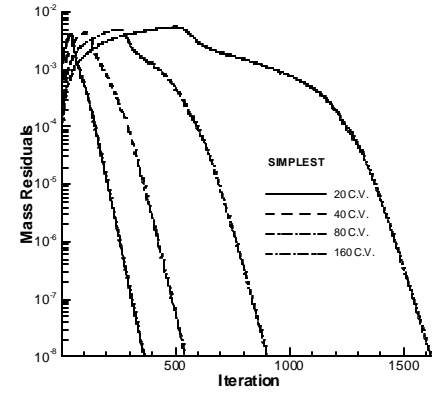
(c)



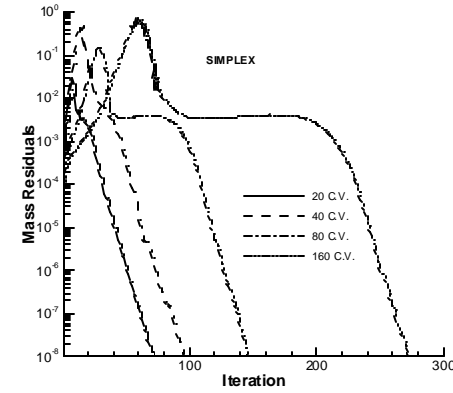
(d)



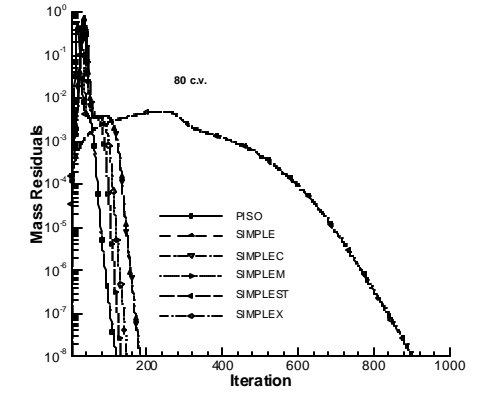
(e)



(f)

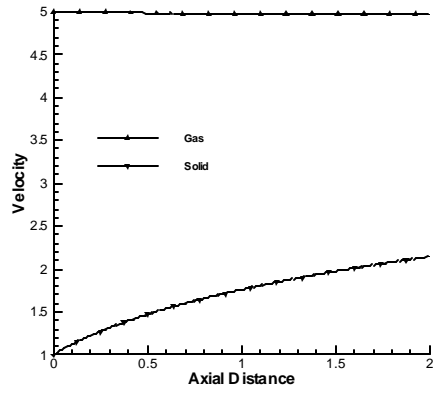


(g)

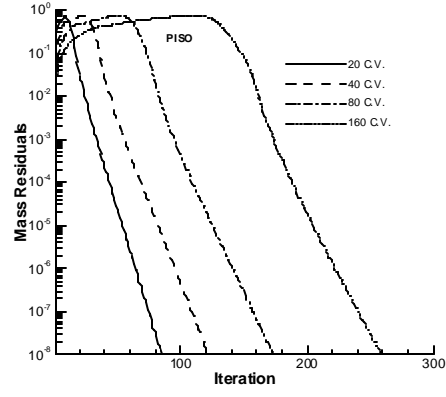


(h)

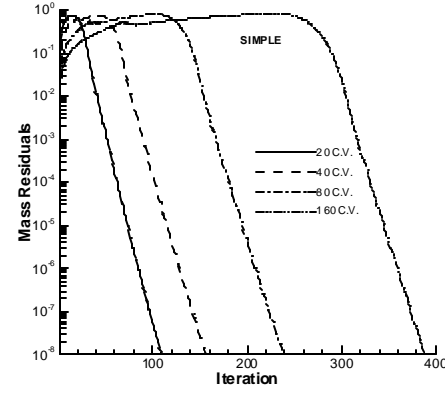
fig. 2 (a) Comparison between the analytical and numerical particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dilute gas-solid flow problem.



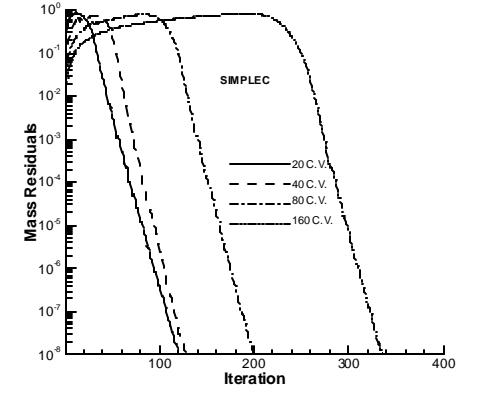
(a)



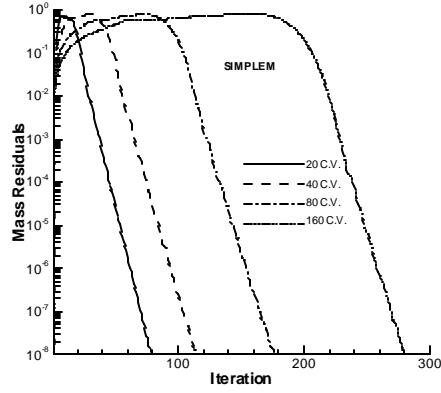
(b)



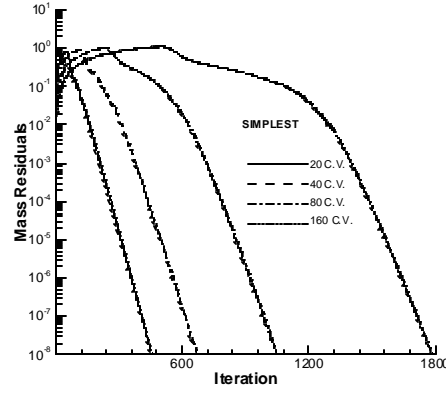
(c)



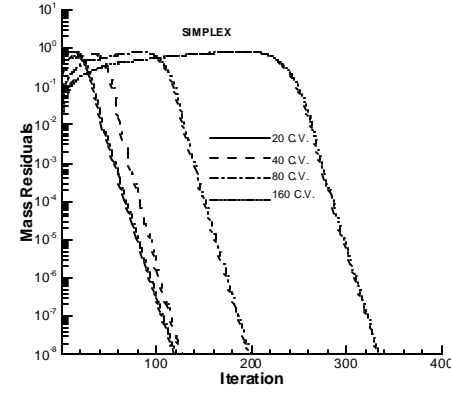
(d)



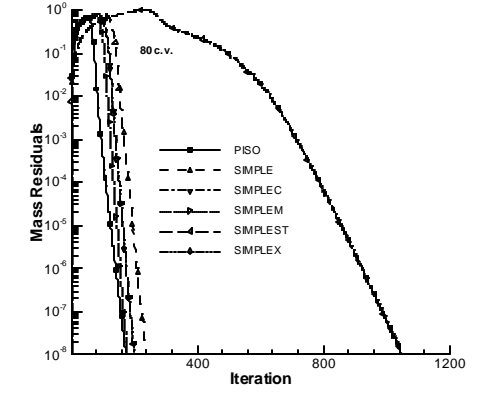
(e)



(f)

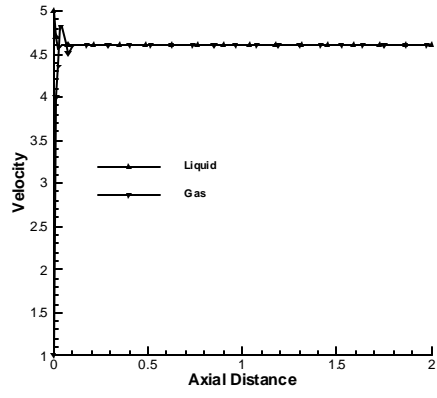


(g)

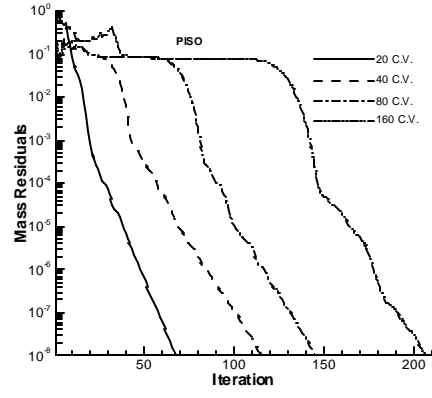


(h)

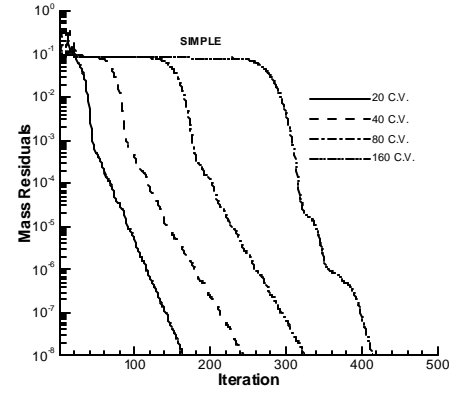
fig. 3 (a) gas and particle velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dense gas-solid flow problem.



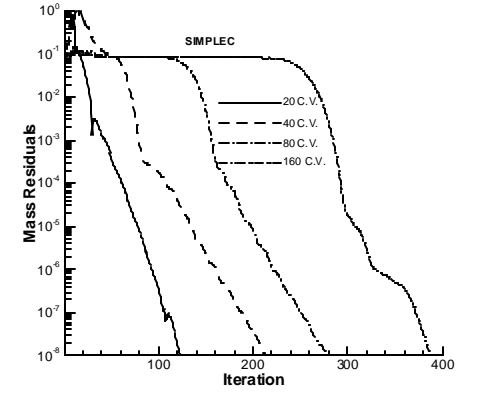
(a)



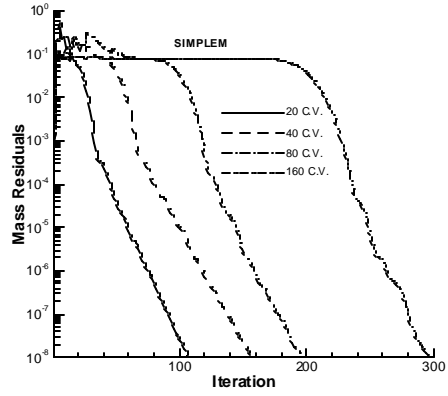
(b)



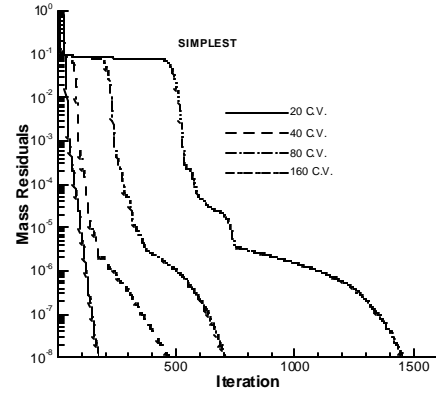
(c)



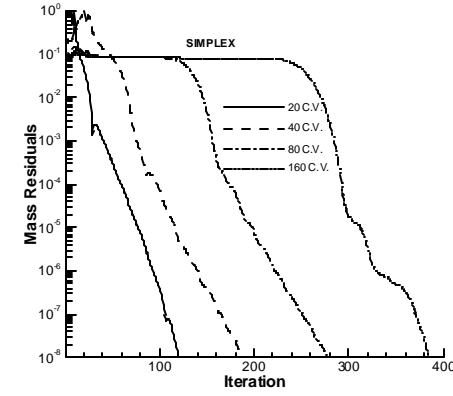
(d)



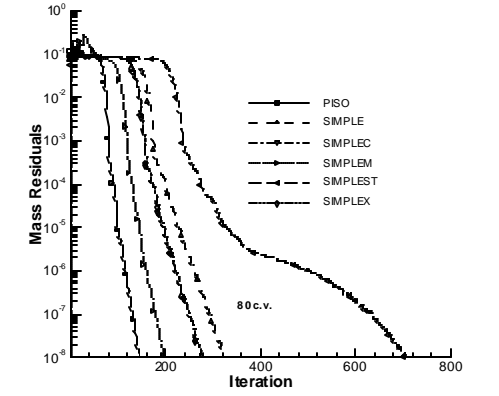
(e)



(f)

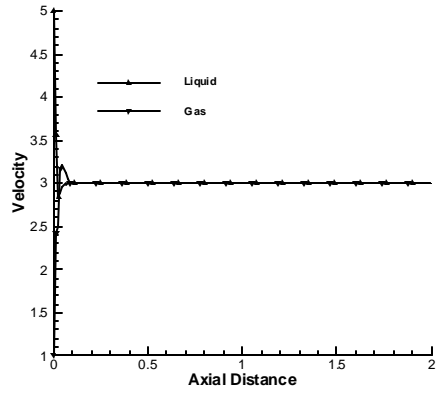


(g)

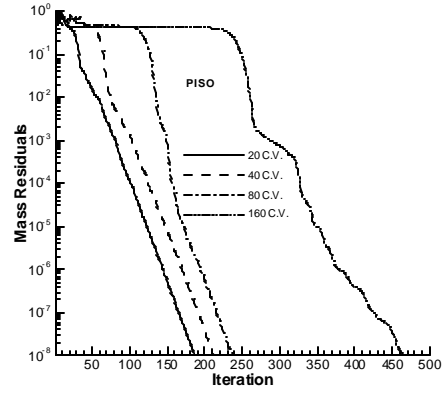


(h)

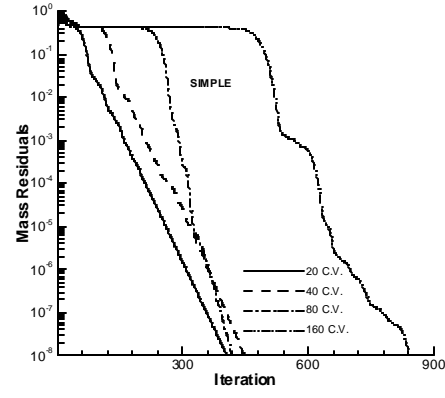
fig. 4 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories of the various algorithms on the 80 C.V. grid for the horizontal dilute bubbly flow problem.



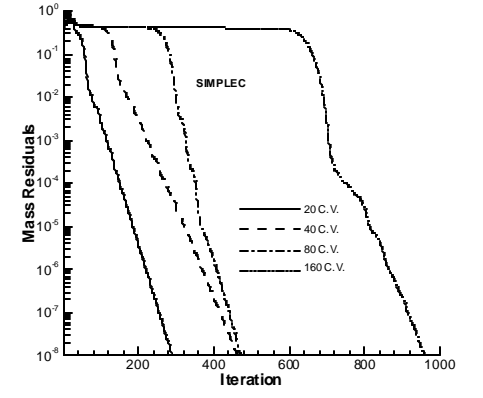
(a)



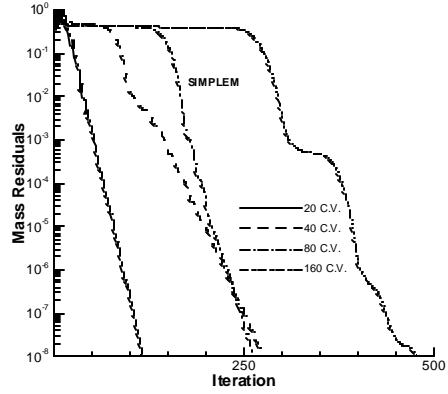
(b)



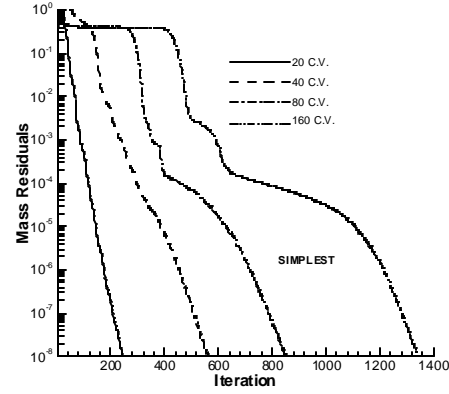
(c)



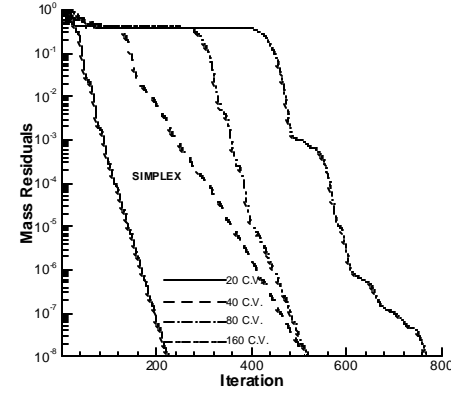
(d)



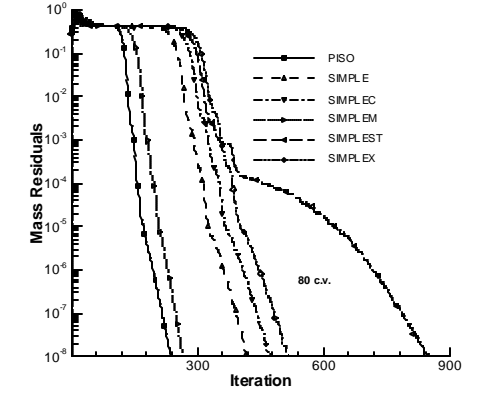
(e)



(f)

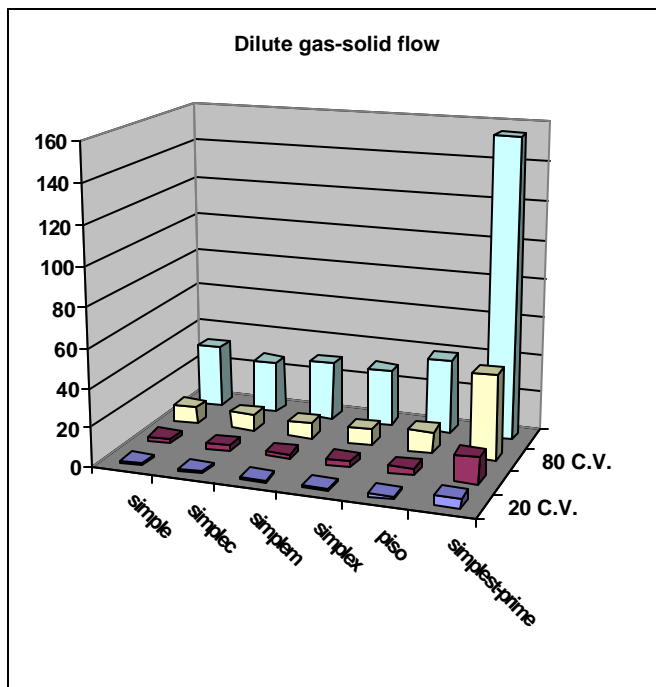


(g)

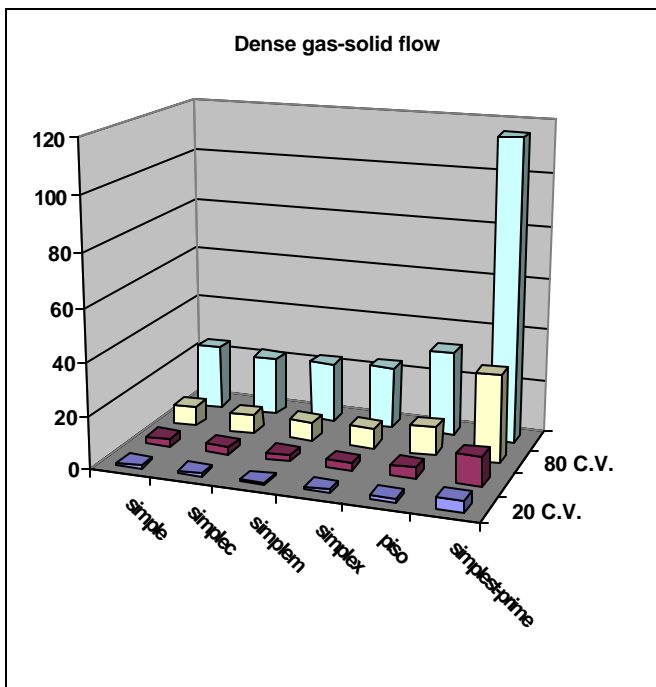


(h)

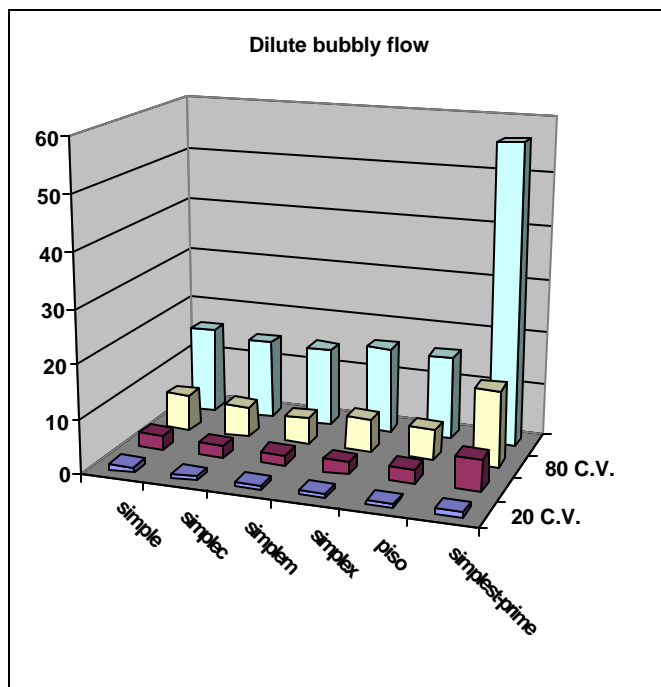
Fig. 5 (a) Liquid and gas velocity distributions, (b)-(g) convergence histories on the different grid systems, (h) and convergence histories on the 80 C.V. grid for the horizontal dense bubbly flow problem.



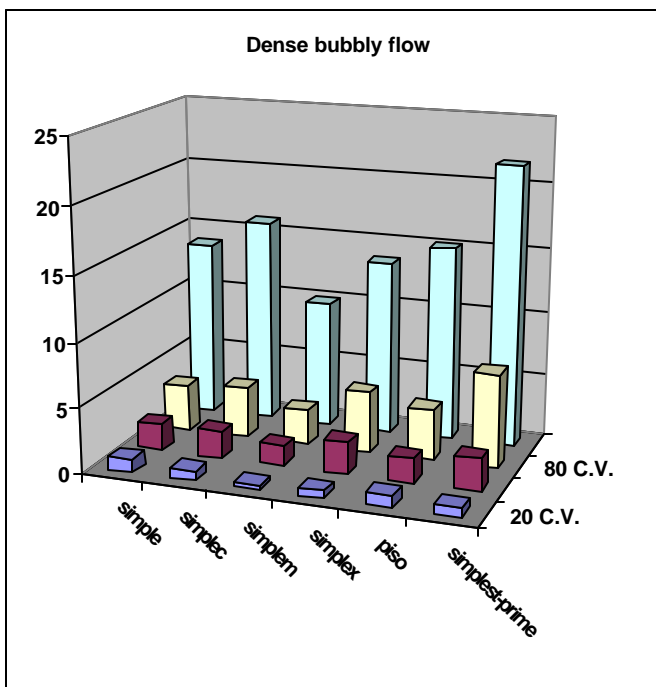
(a)



(b)

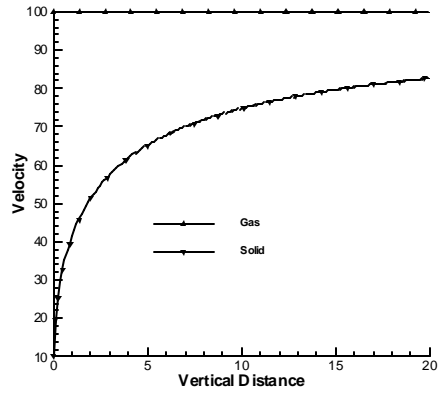


(c)

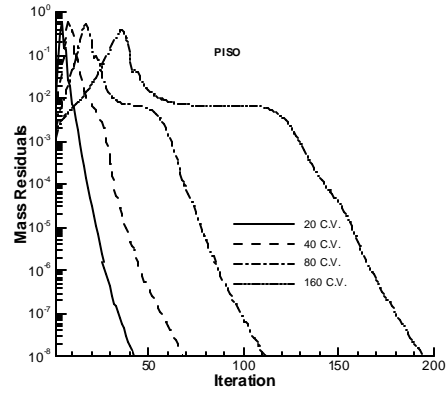


(d)

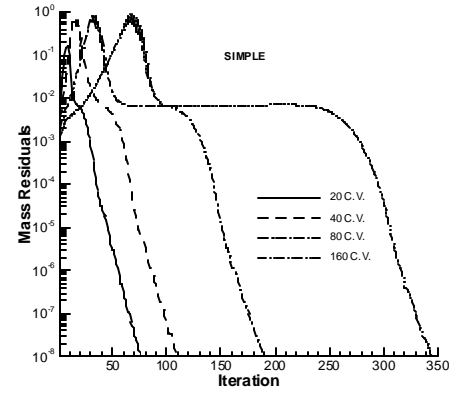
Fig. 6 Normalized CPU-times for the horizontal (a) dilute gas-solid, (b) dense gas-solid, (c) dilute bubbly, and (d) dense bubbly flow problem.



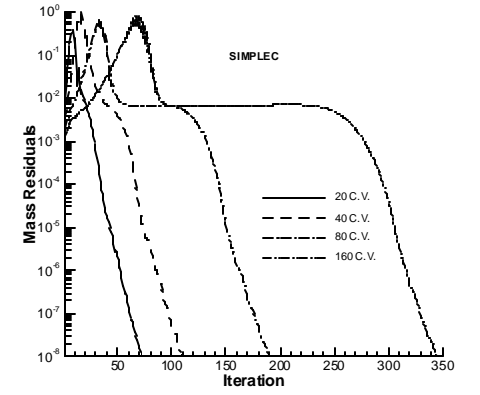
(a)



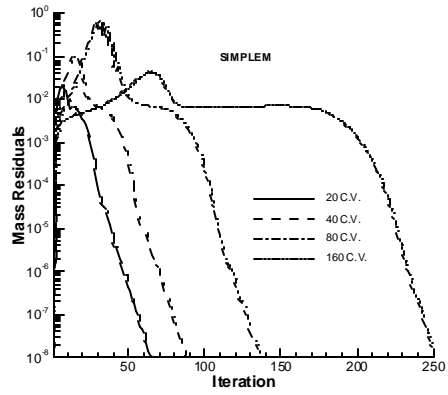
(b)



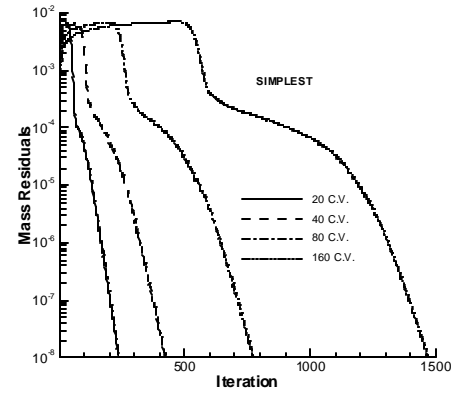
(c)



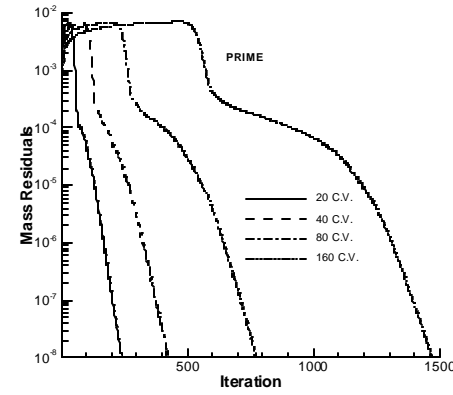
(d)



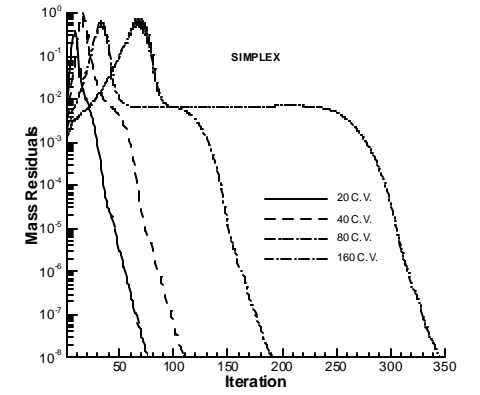
(e)



(f)

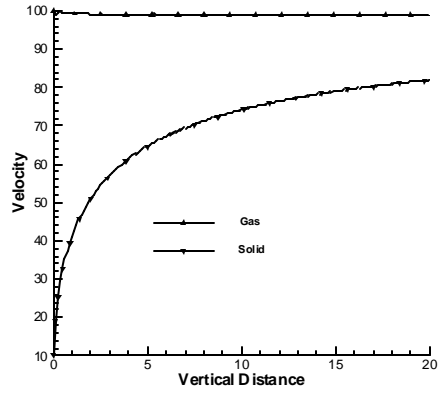


(g)

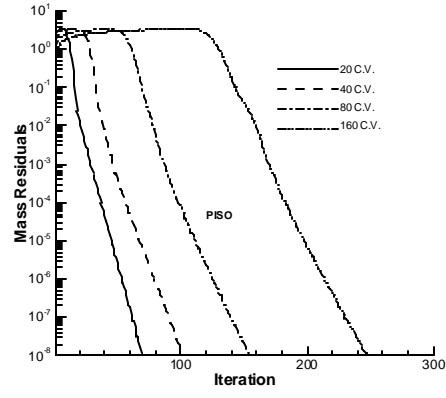


(h)

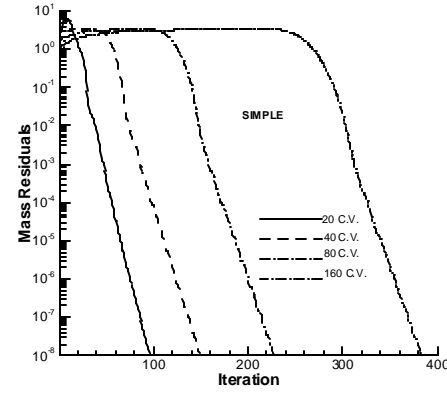
fig. 7 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dilute gas-solid flow problem.



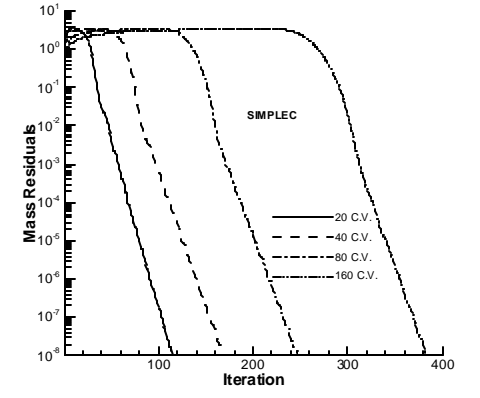
(a)



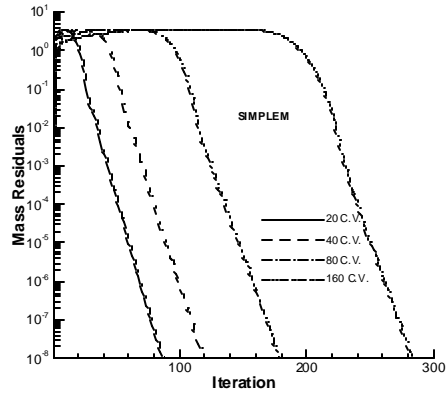
(b)



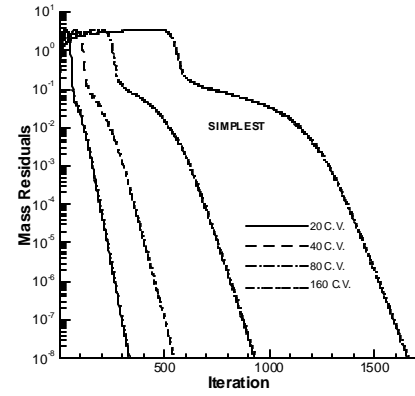
(c)



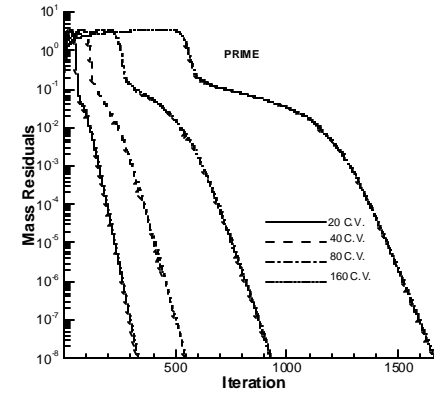
(d)



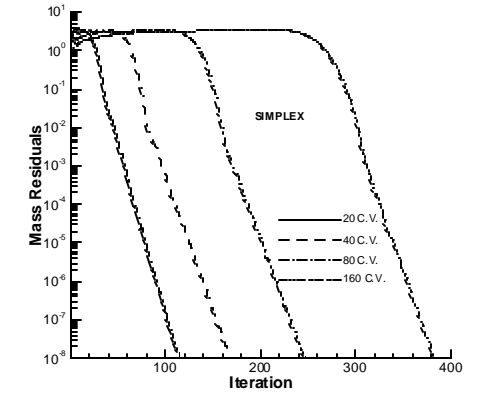
(e)



(f)

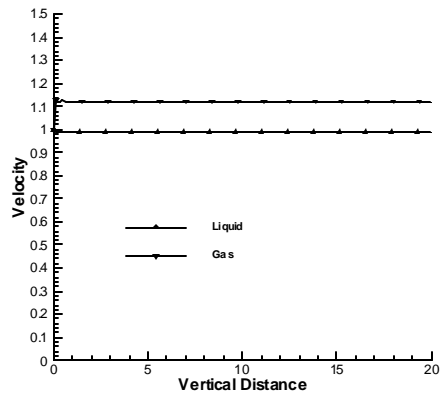


(g)

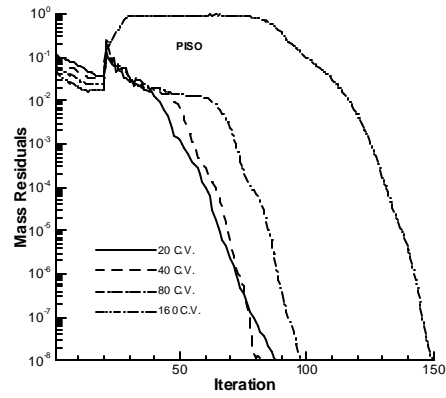


(h)

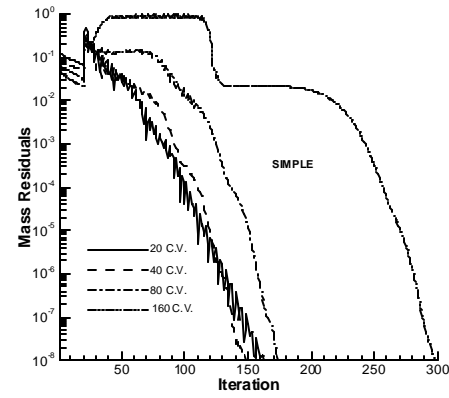
fig. 8 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense gas-solid flow problem.



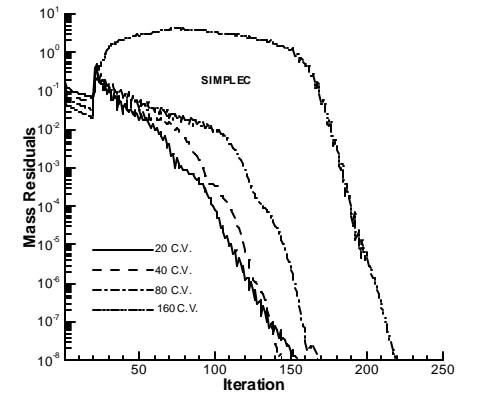
(a)



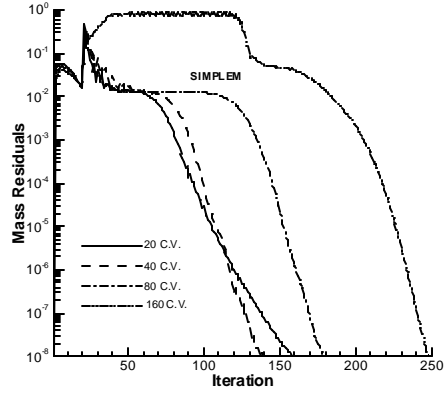
(b)



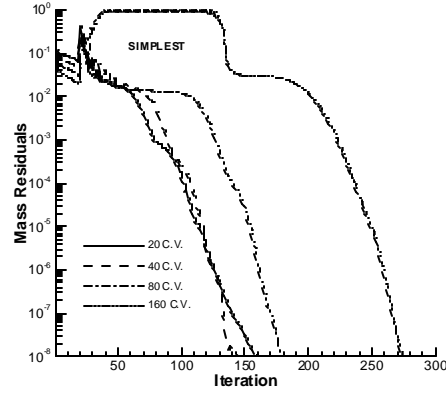
(c)



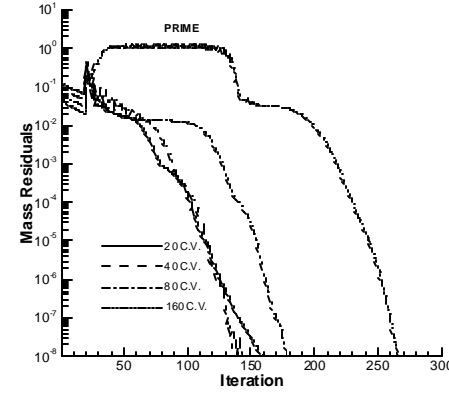
(d)



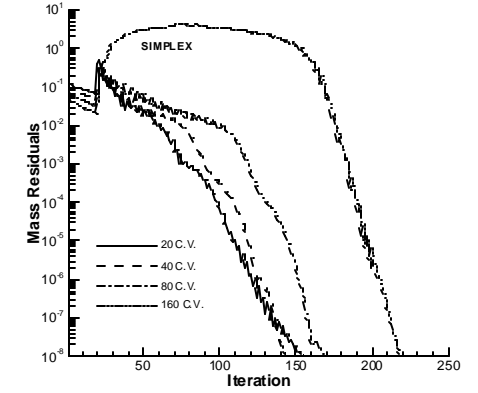
(e)



(f)

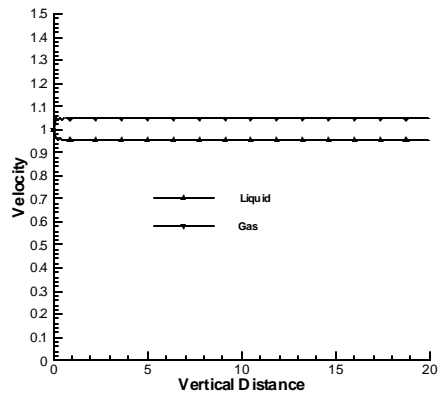


(g)

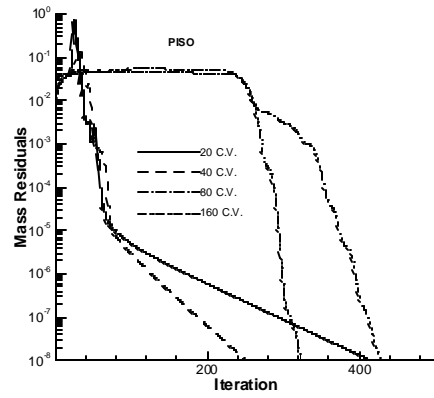


(h)

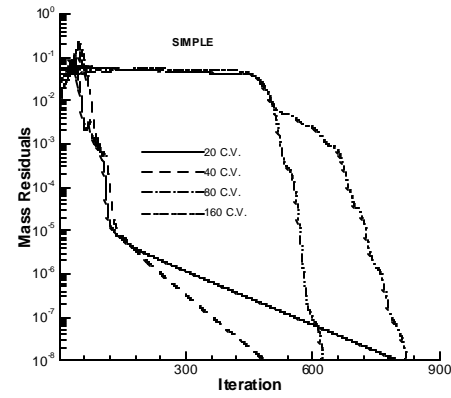
fig. 9 (a) gas and particle velocity distributions; and (b)-(h) convergence histories on the different grid systems for the vertical dilute bubbly flow problem.



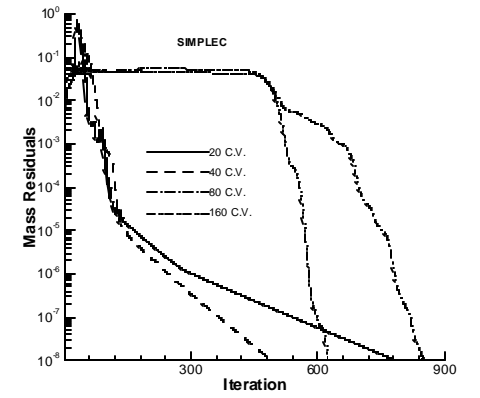
(a)



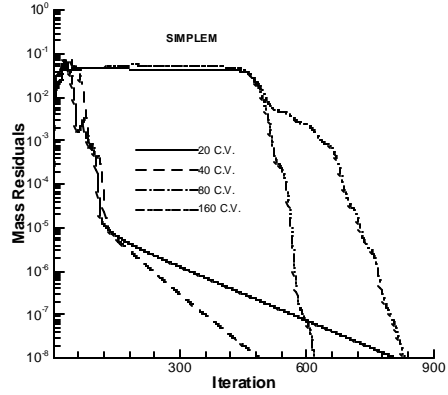
(b)



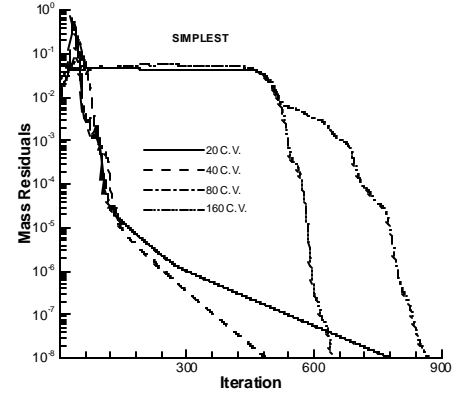
(c)



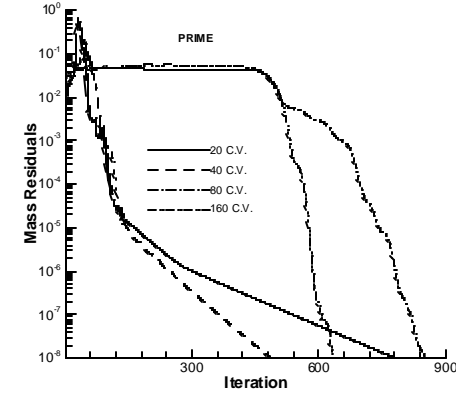
(d)



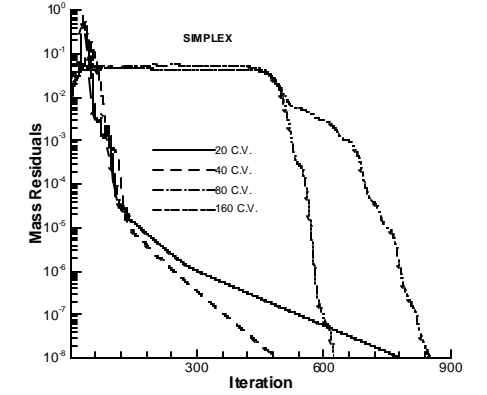
(e)



(f)

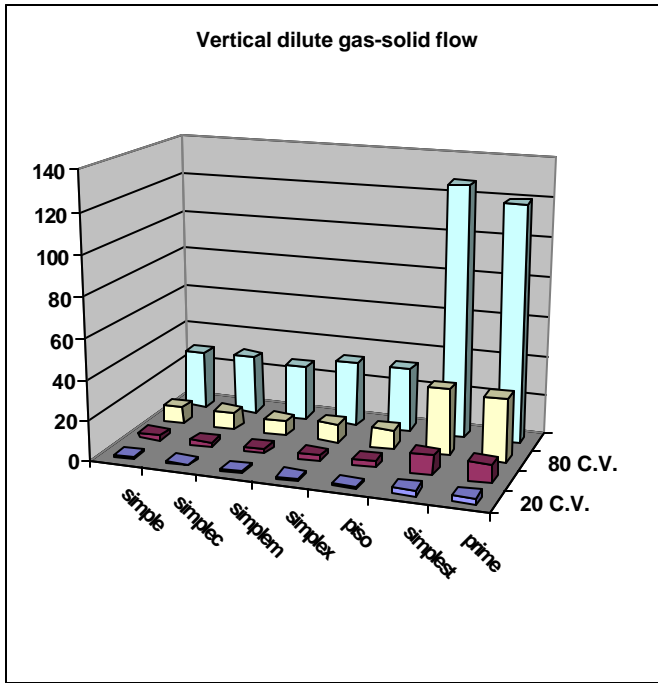


(g)

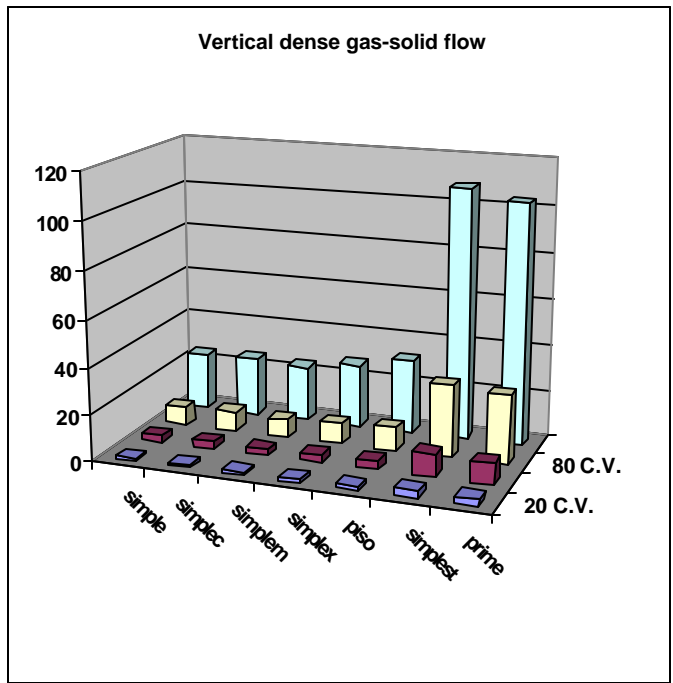


(h)

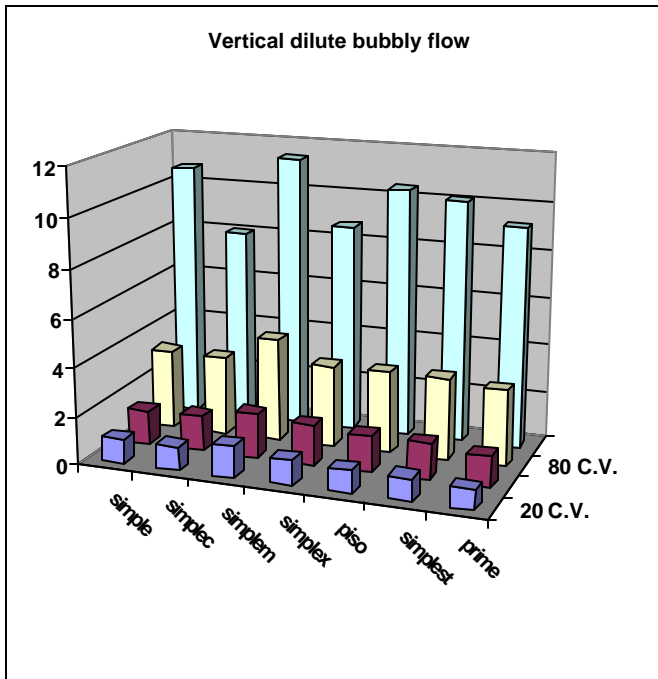
fig. 10 (a) gas and particle velocity distributions, and (b)-(h) convergence histories on the different grid systems for the vertical dense bubbly flow problem.



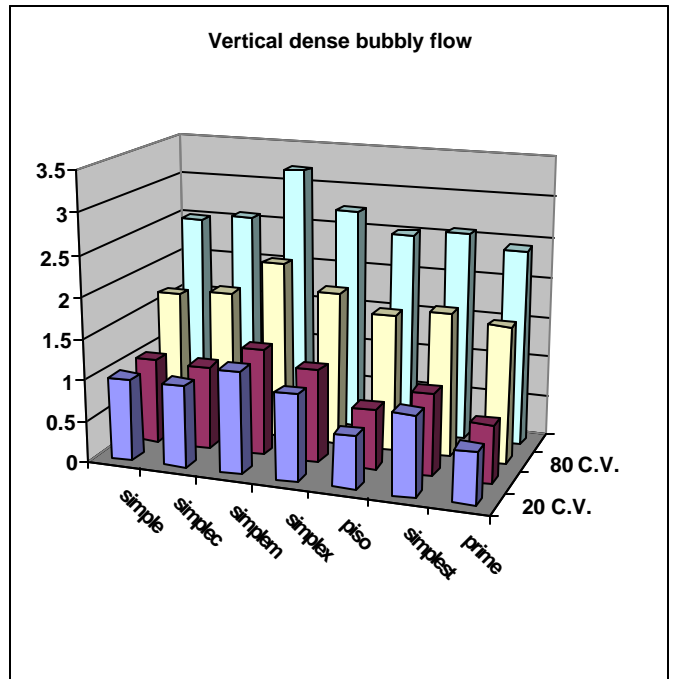
(a)



(b)



(c)



(d)

Fig. 11 Normalized CPU-times for the vertical (a) dilute gas-solid, (b) dense gas-solid, (c) dilute bubbly, and (d) dense bubbly flow problem.

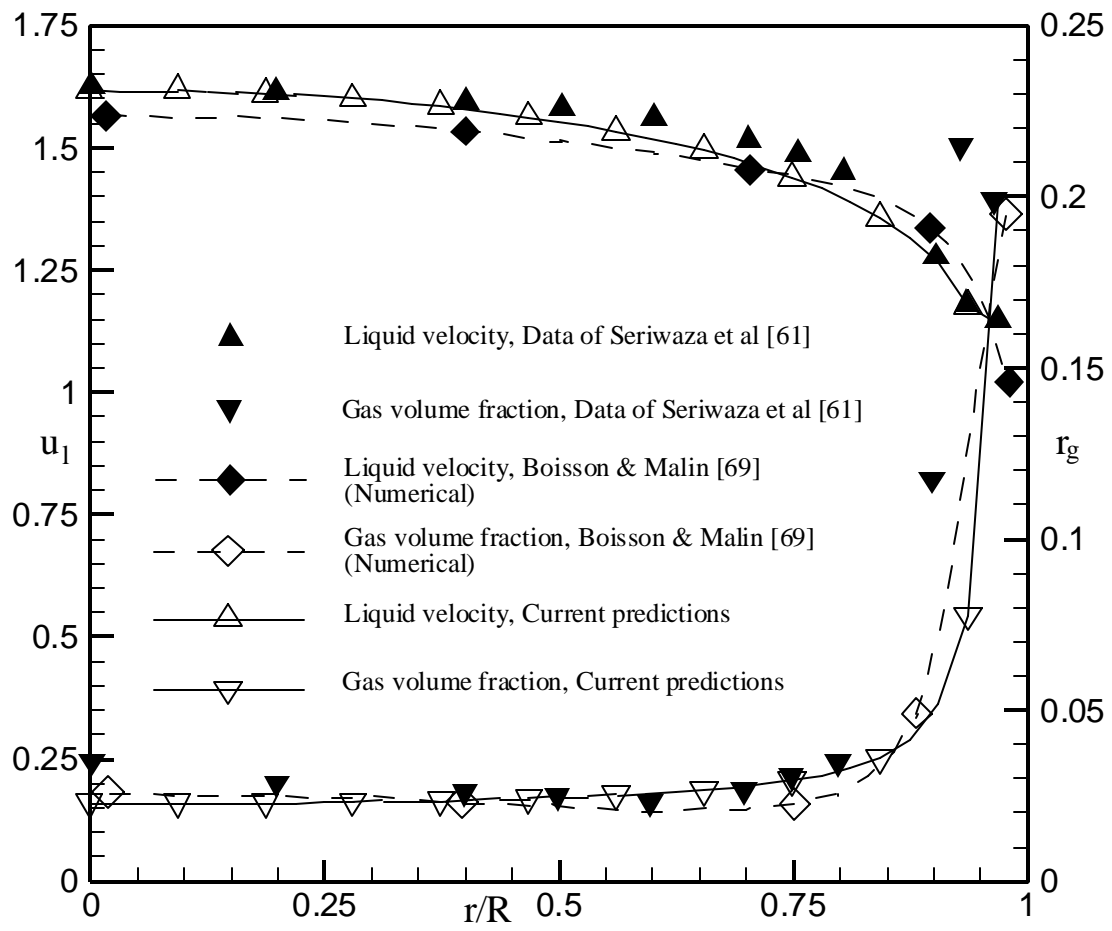


Fig. 12 Comparison of fully developed liquid velocity and void fraction profiles for turbulent upward bubbly flow in a pipe.

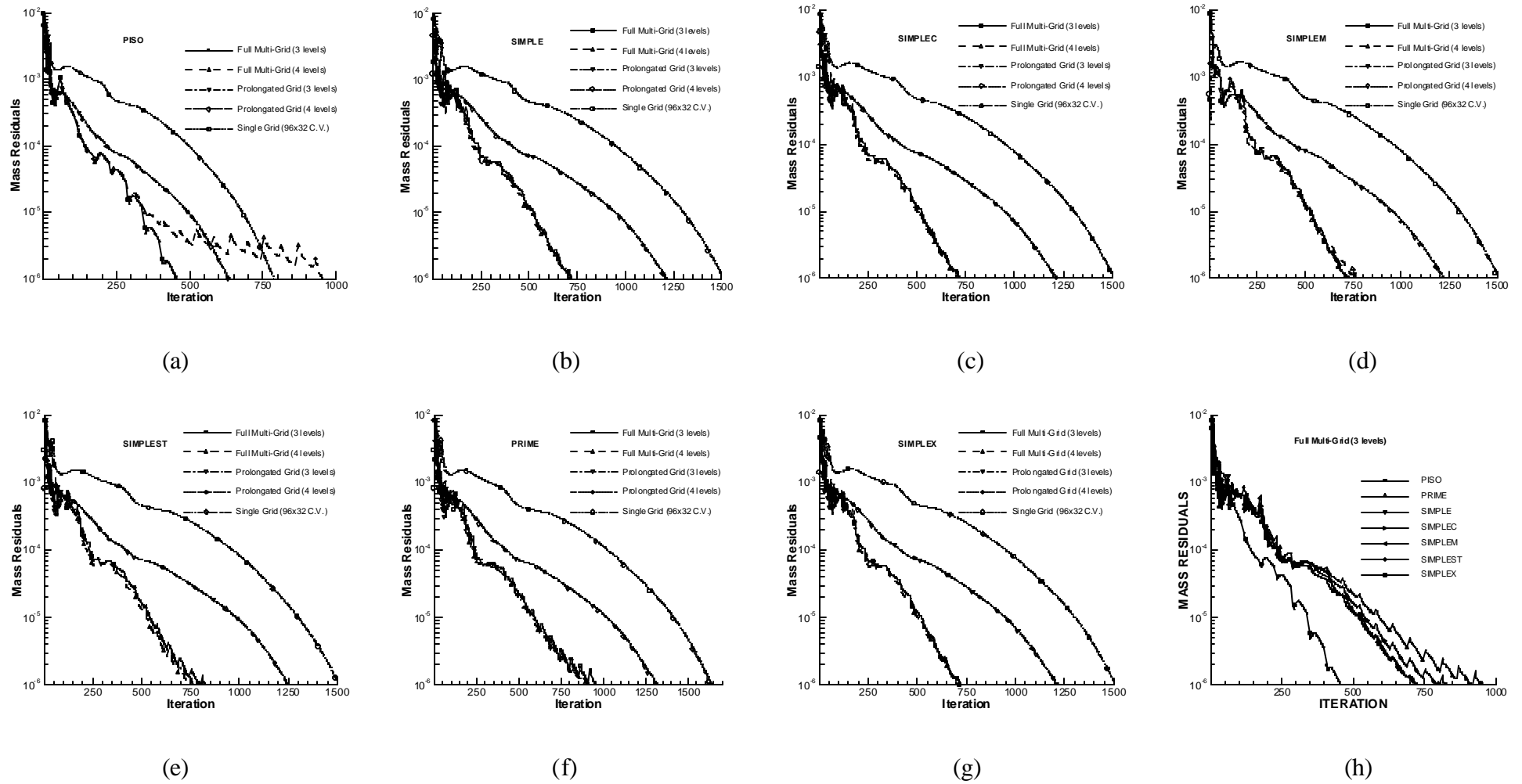


fig. 13 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for turbulent upward bubbly flow in a pipe.

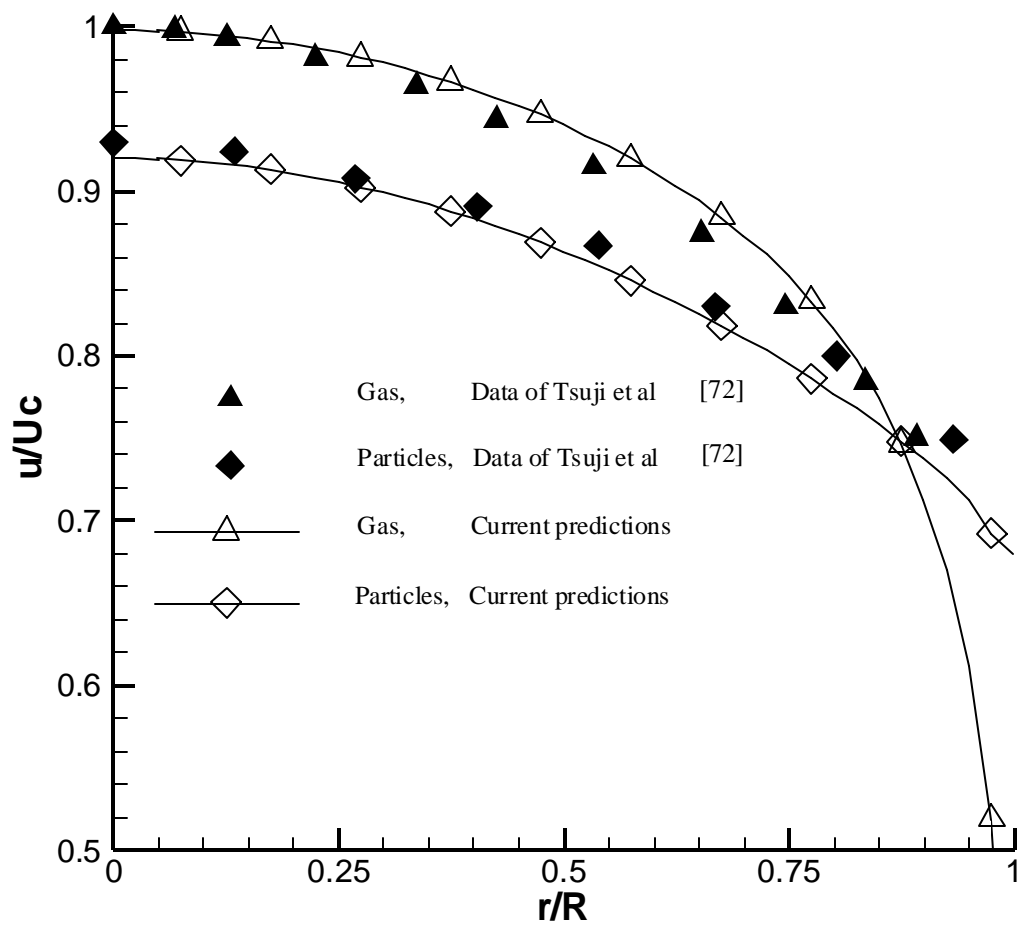
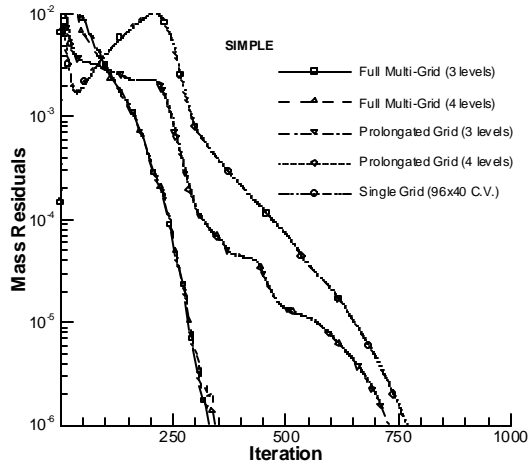
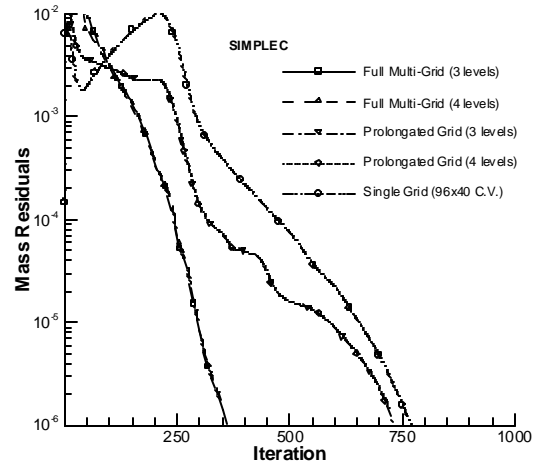


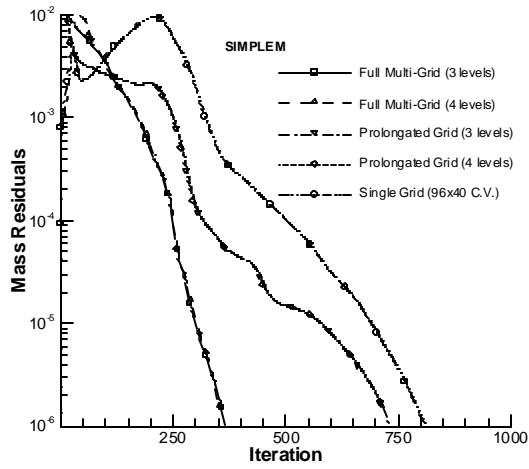
Fig. 14 Comparison of fully developed gas and particle velocity profiles for turbulent air-particle flow in a pipe.



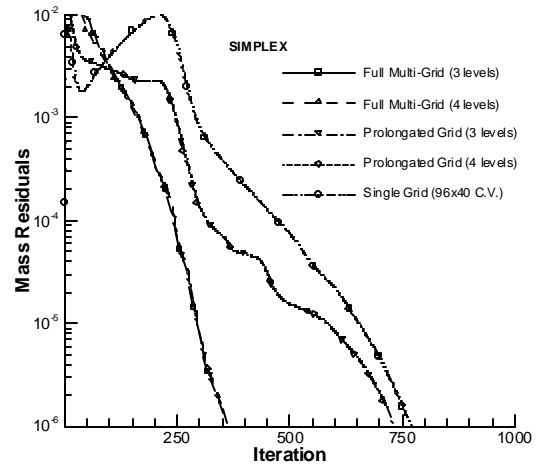
(a)



(b)



(c)



(d)

Fig. 15 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEM, and (d) SIMPLEX algorithms using the SG, PG, and FMG methods on the finest mesh for turbulent air-particle flow in a pipe.

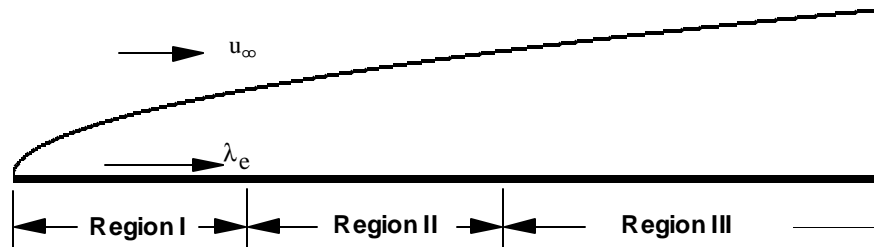


Fig. 16 The three different regions within the boundary layer of dusty flow over a flat plate.

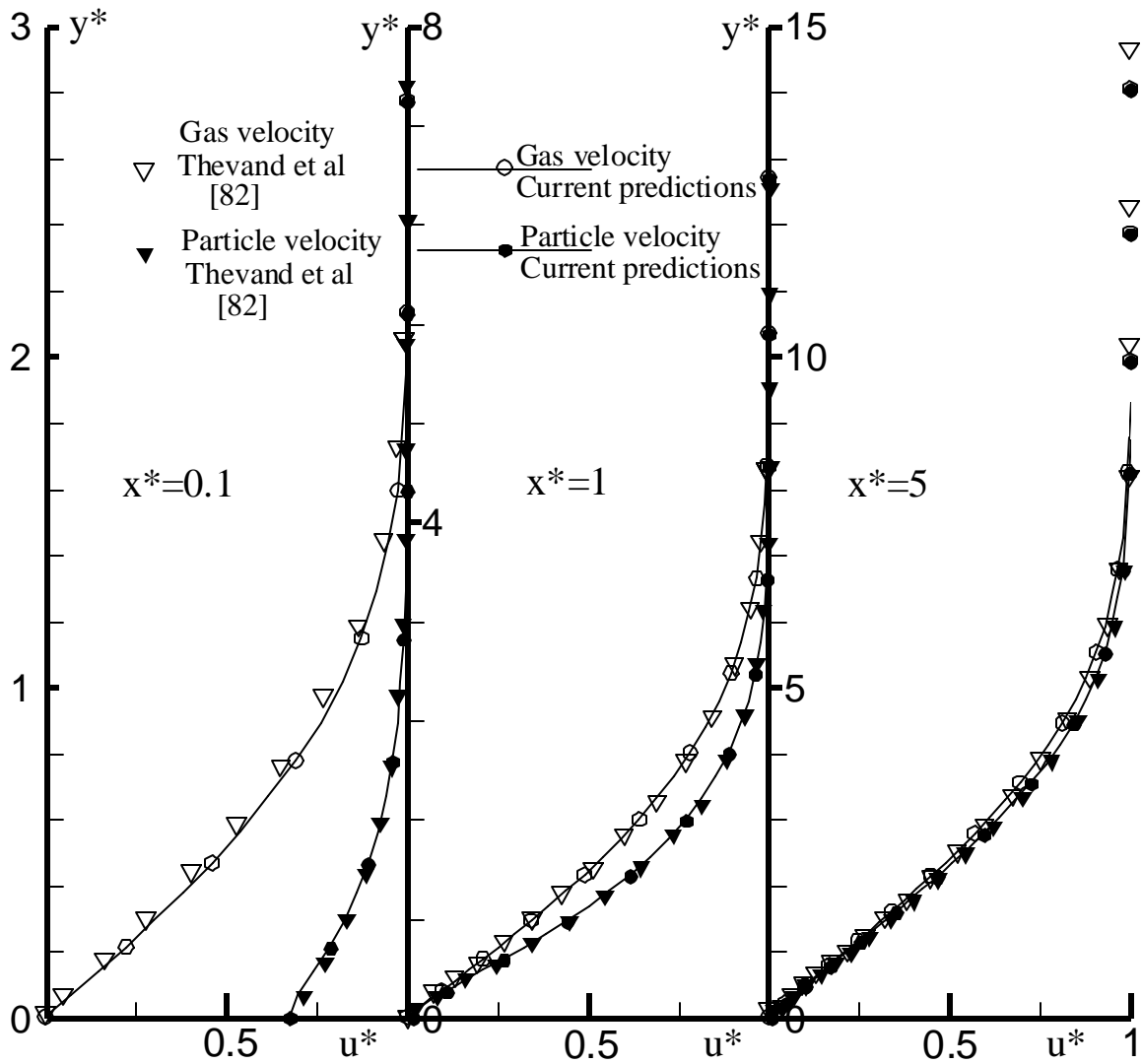
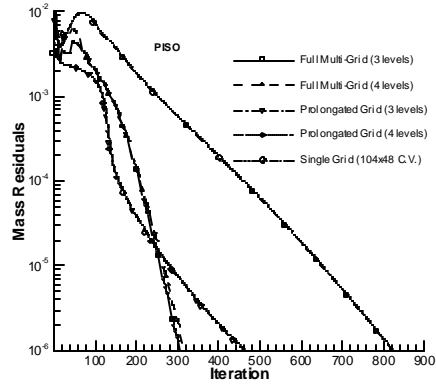
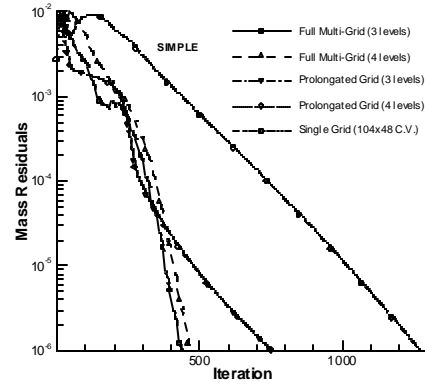


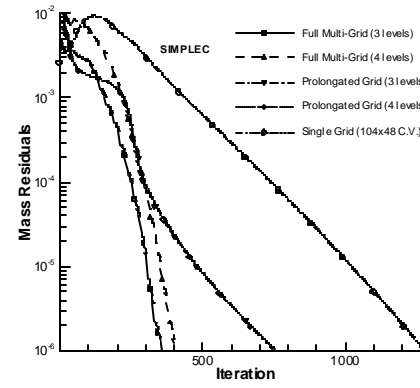
Fig. 17 Comparison of fully developed gas and particle velocity profiles inside the boundary layer at different axial locations for dilute two-phase flow over a flat plate.



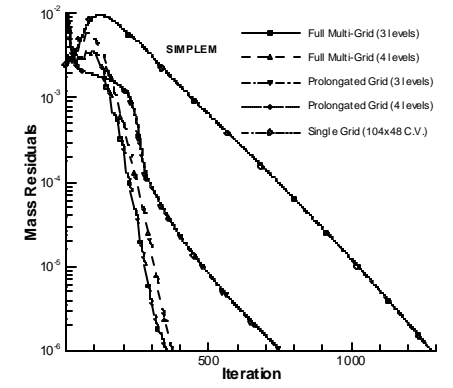
(a)



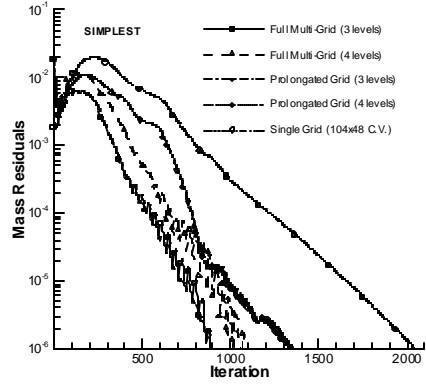
(b)



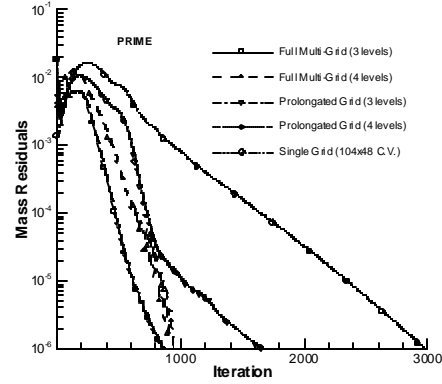
(c)



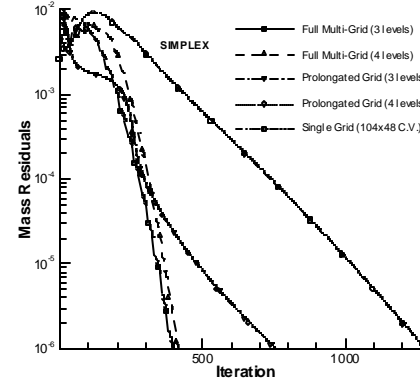
(d)



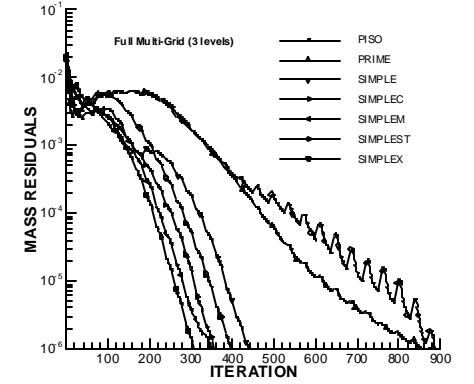
(e)



(f)



(g)



(h)

Fig. 18 (a)-(g) Convergence histories of the SG, PG, and FMG methods on the finest grid, and (h) convergence histories of the various algorithms on the finest mesh using the FMG method for dusty gas flow over a flat plate.

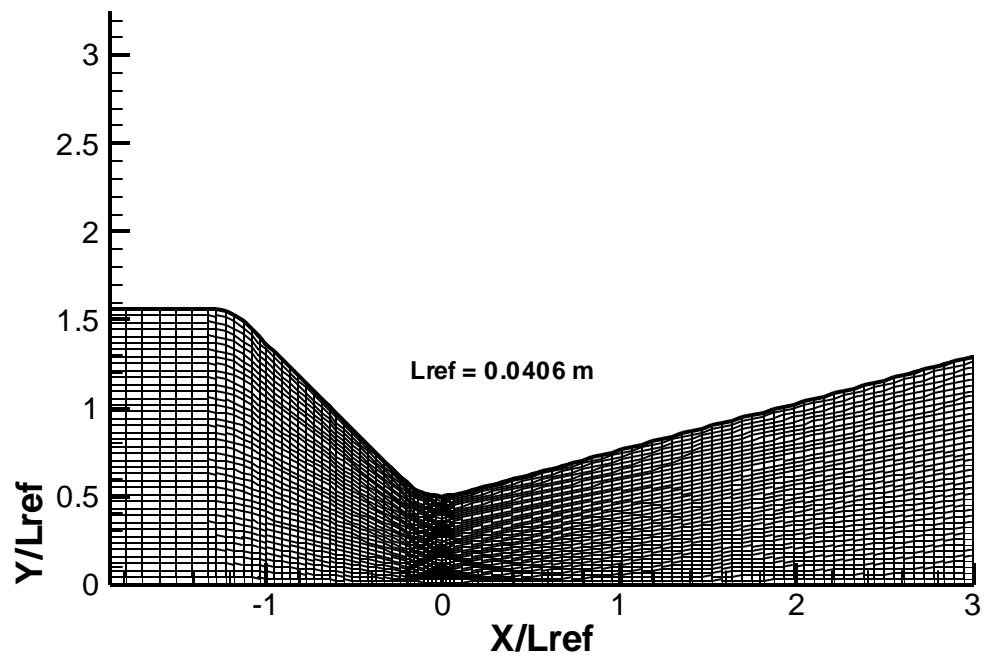
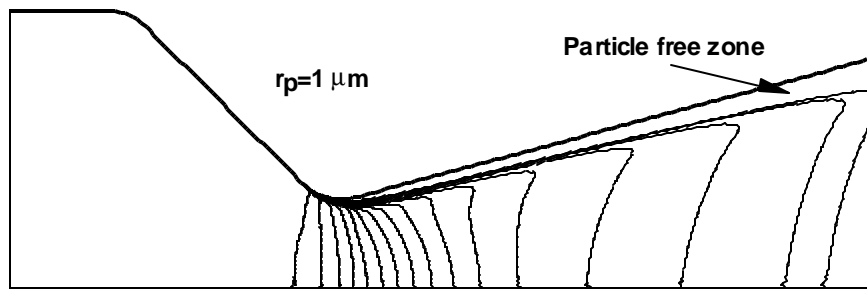
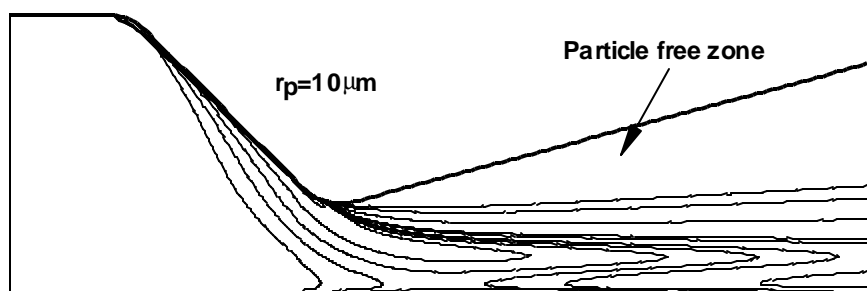


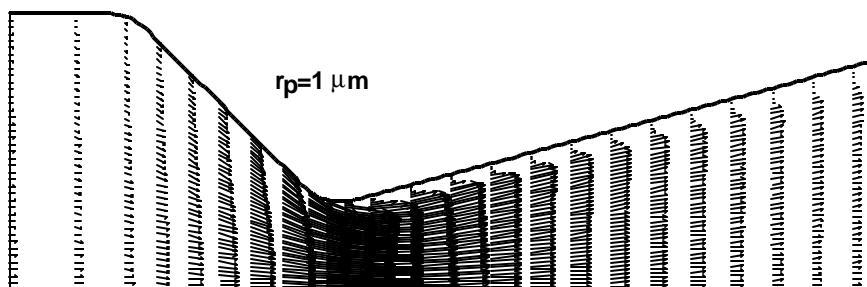
Fig. 19 Physical domain for the dusty gas flow in a converging-diverging nozzle.



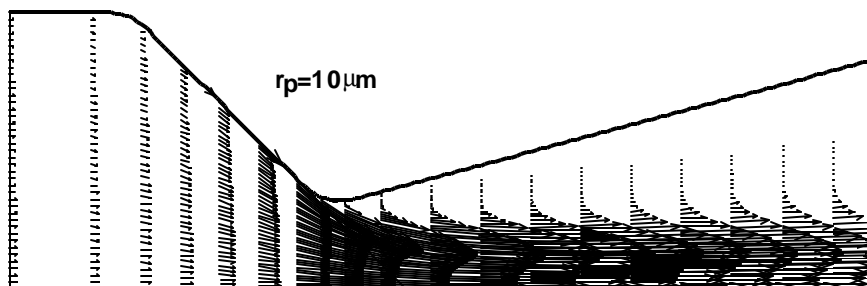
(a)



(b)

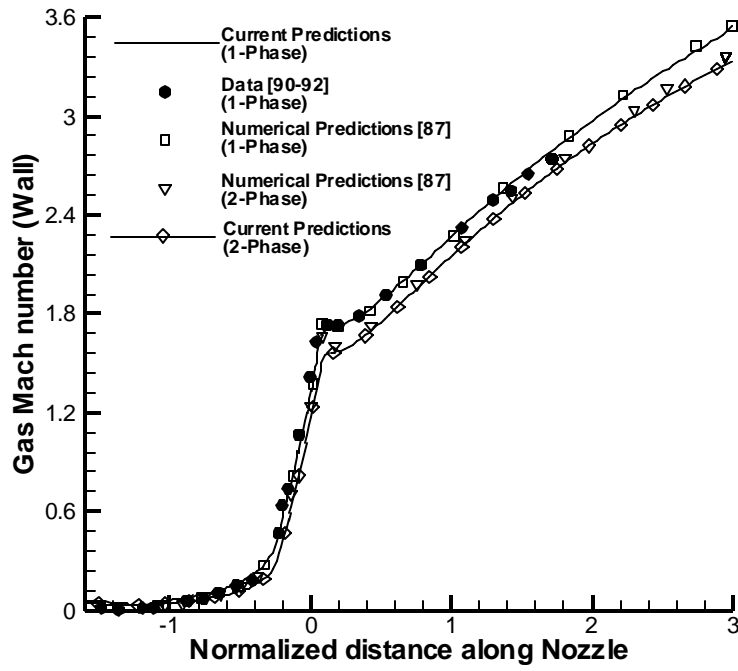


(c)

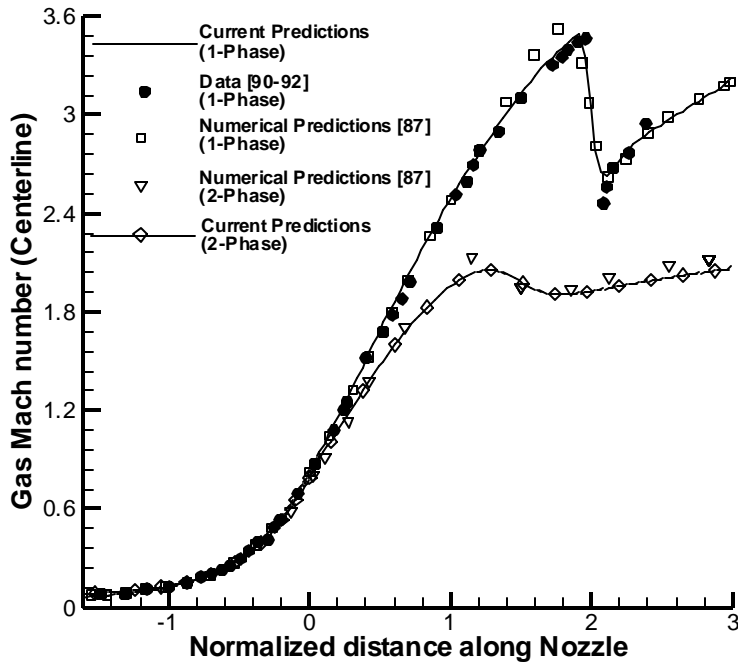


(d)

Fig. 20 (a,b) Volume Fraction contours and (c,d) particle velocity vectors for dusty gas flow in a converging-diverging nozzle.

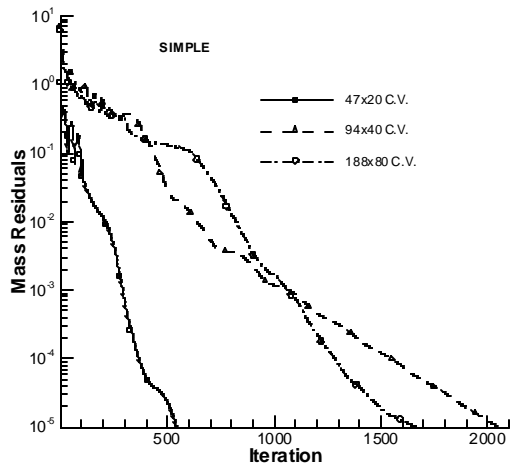


(a)

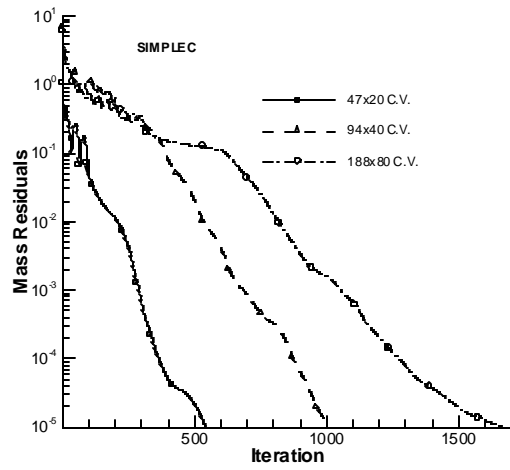


(b)

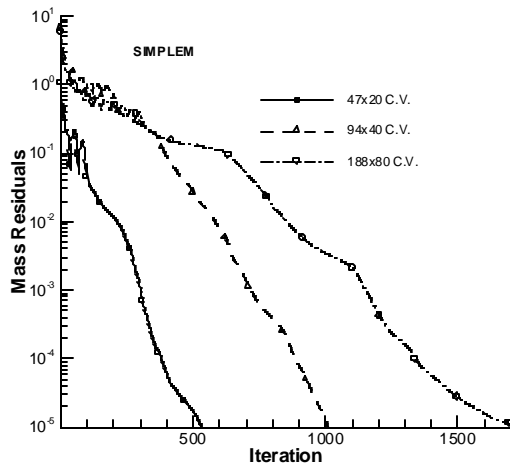
Fig. 21 Comparison of one-phase and two-phase gas Mach number distributions along the (a) wall and (b) centerline of the dusty flow in a converging-diverging nozzle problem.



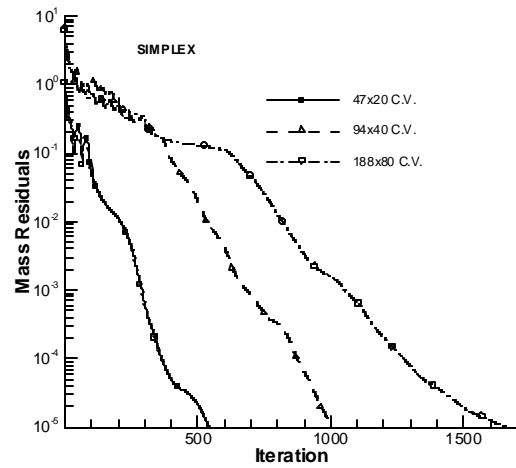
(a)



(b)



(c)



(d)

Fig. 22 Convergence histories of the (a) SIMPLE, (b) SIMPLEC, (c) SIMPLEM, and (d) SIMPLEX algorithms using the PG method for dusty gas flow in a converging-diverging nozzle.

Table 1 Normalized CPU-times for turbulent bubbly flow in a pipe.

GRID	METHOD	ALGORITHMS						
		SIMPLE	SIMPLEC	SIMPLEX	SIMPLEST	SIMPLEM	PISO	PRIME
48x16 C.V.	SG	1.00	1.00	1.02	1.04	1.09	1.08	1.19
96x32 C.V.	SG	40.09	40.42	40.98	40.66	43.07	41.82	43.59
	PG (3 levels)	34.24	34.41	35.02	36.82	37.12	34.48	36.05
	PG (4 levels)	34.34	34.25	34.67	36.88	36.93	34.50	36.07
	FMG (3 levels)	22.32	22.17	22.40	25.55	23.67	27.31	28.38
	FMG (4 levels)	22.53	22.51	22.78	23.81	25.55	55.68	27.89

Table 2 Normalized CPU-times for turbulent air-particle flow in a pipe.

		ALGORITHMS			
GRID	METHOD	SIMPLE	SIMPLEC	SIMPLEX	SIMPLEM
48x20 C.V.	SG	1.00	1.26	1.28	1.45
96x40 C.V.	SG	18.81	18.85	19.07	21.05
	PG (3 levels)	18.94	19.37	19.37	20.22
	PG (4 levels)	19.03	19.46	19.44	20.30
	FMG (3 levels)	11.11	12.34	12.44	13.72
	FMG (4 levels)	12.07	13.10	13.25	13.54

Table 3 Normalized CPU-times for Dusty flow over a flat plate.

		ALGORITHMS						
GRID	METHOD	SIMPLE	SIMPLEC	SIMPLEX	SIMPLEST	SIMPLEM	PISO	PRIME
52x24 C.V.	SG	1.00	1.04	1.11	1.52	1.31	1.12	2.13
104x48 C.V.	SG	16.92	18.09	18.94	26.00	21.48	19.46	45.87
	PG (3 levels)	14.25	14.94	15.51	18.85	17.48	15.84	24.48
	PG (4 levels)	14.27	14.80	15.50	18.90	17.56	15.79	24.46
	FMG (3 levels)	5.35	5.44	5.88	11.87	7.84	8.34	12.19
	FMG (4 levels)	6.12	6.14	6.66	12.98	6.66	9.08	9.98

Table 4 Normalized CPU-times for Dusty flow in a converging-diverging nozzle.

GRID	METHOD	ALGORITHMS			
		SIMPLE	SIMPLEC	SIMPLEX	SIMPLEM
47x20 C.V.	SG	1.00	1.02	1.06	1.13
94x40 C.V.	PG (2 levels)	13.08	9.24	9.47	11.37
188x80 C.V.	PG (3 levels)	82.43	88.49	91.35	117.83